



Universidade de Brasília - UnB
Faculdade UnB Gama - FGA
Engenharia de Software

Evoluindo uma plataforma de rede de colaboração social para um ambiente de aprendizagem virtual

Autor: Hebert Douglas de Almeida Santos
Orientador: Prof. Dr. Paulo Roberto Miranda Meirelles

Brasília, DF
2015



Hebert Douglas de Almeida Santos

Evoluindo uma plataforma de rede de colaboração social para um ambiente de aprendizagem virtual

Monografia submetida ao curso de graduação em (Engenharia de Software) da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em (Engenharia de Software).

Universidade de Brasília - UnB

Faculdade UnB Gama - FGA

Orientador: Prof. Dr. Paulo Roberto Miranda Meirelles

Brasília, DF

2015

Hebert Douglas de Almeida Santos

Evoluindo uma plataforma de rede de colaboração social para um ambiente de aprendizagem virtual/ Hebert Douglas de Almeida Santos. – Brasília, DF, 2015-
91 p. : il. (algumas color.) ; 30 cm.

Orientador: Prof. Dr. Paulo Roberto Miranda Meirelles

Trabalho de Conclusão de Curso – Universidade de Brasília - UnB
Faculdade UnB Gama - FGA , 2015.

1. Redes sociais. 2. Software livre. I. Prof. Dr. Paulo Roberto Miranda Meirelles. II. Universidade de Brasília. III. Faculdade UnB Gama. IV. Evoluindo uma plataforma de rede de colaboração social para um ambiente de aprendizagem virtual

CDU 02:141:005.6

Hebert Douglas de Almeida Santos

Evoluindo uma plataforma de rede de colaboração social para um ambiente de aprendizagem virtual

Monografia submetida ao curso de graduação em (Engenharia de Software) da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em (Engenharia de Software).

Trabalho aprovado. Brasília, DF, 15 de Dezembro de 2015:

**Prof. Dr. Paulo Roberto Miranda
Meirelles**
Orientador

Msc. Hilmer Rodrigues Neri
Convidado 1

Msc. Joênio Costa
Convidado 2

Brasília, DF
2015

Agradecimentos

Agradeço primeiramente a Deus, por ter permitido seguir estudando e trabalhando com saúde e sempre me dando forças para superar as dificuldades.

Aos meus pais, Donizete dos Santos e Ednea de Almeida Santos, pela educação que me foi dada ao longo da vida, pelo esforço que fizeram para permitir que estude sem preocupações e por sempre terem me incentivado a estudar para ser uma pessoa melhor. Agradeço também ao meu irmão, Albert de Almeida Santos, por me dar forças e estar disposto à ajudar sempre que necessário.

Ao Prof. Dr. Paulo Meirelles pela orientação ao longo deste trabalho; pelas oportunidades profissionais compartilhadas; por todos os ensinamentos técnicos; principalmente na disciplina de manutenção e evolução que mostrou a importância da contribuição e trabalho em equipe, além de permitir a colaboração com softwares livres em produção. Me propiciou fatores fundamentais para o meu crescimento profissional.

Aos membros do Laboratório Avançado de Produção, Pesquisa e Inovação em Software - LAPPIS por proporcionar um ambiente amigável e colaborativo onde sempre há um crescimento em conjunto graças ao trabalho em equipe.

Aos meus amigos e colegas de infância, e aqueles que conheci ao longo dessa jornada que auxiliaram, motivaram e dividiram momentos de dificuldades e felicidades ao meu lado.

*“Ensinar não é transferir conhecimento,
mas criar as possibilidades para a sua própria
produção ou a sua construção.”
(Paulo Freire)*

Resumo

Este trabalho de conclusão de curso de engenharia de software tem como objetivo evoluir uma plataforma de redes sociais, para permitir que ele contenha funcionalidades de um Ambiente Virtual de Aprendizagem (AVA), propondo um hibridismo entre esses ambientes. Neste contexto, escolheu-se utilizar a rede colaboração livre Comunidade.UnB, um ambiente para alunos e professores trabalharem de maneira horizontal e colaborativa, que faz uso da plataforma brasileira para redes sociais livre Noosfero. Dessa maneira, o estudo contemplou um levantamento sobre os principais recursos desses ambientes de aprendizagem comparando-os com o Noosfero, proporcionando a implementação de um conjunto de funcionalidades e melhorias, de maneira que atendessem as necessidades levantadas. Este estudo evidencia a utilização da arquitetura de *plugins* proposta pela plataforma em questão, além de compreender como realizar a evolução da plataforma no desenvolvimento empírico de software. Além disso foi explanado o processo de desenvolvimento colaborativo em uma comunidade de software livre, compartilhando o conhecimento adquirido com uma equipe de desenvolvedores na UnB Gama.

Palavras-chaves: redes sociais. ambiente virtual de aprendizagem. noosfero. métodos ágeis. software livre.

Abstract

This study of Software Engineering aims to develop a platform of social networking, with the scope to allow the addition of features of a Virtual Learning Environment (VLE), proposing a hybridism between these environments. In this context, the free collaboration network Comunidade.UnB was chosen, an environment for students and teachers to work in horizontal and collaborative way, making use of Brazilian platform for free social networks, Noosfero. Thus, the study included a survey of the main features of these learning environments by comparing them with Noosfero, providing the implementation of a set of features and improvements, in order to solve the needs raised. This study shows the use of the proposed plugin architecture for the platform in question, and understand how to perform the evolution of the platform in the empirical development of software. Also explains the collaborative development process in an open source community, sharing the knowledge gained with a team of developers at UnB Gama.

Key-words: social networking. virtual learning environment. noosfero. agile methods. open-source software.

Lista de ilustrações

Figura 1 – Issue Tracker no GitLab	37
Figura 2 – Arquitetura do Noosfero	38
Figura 3 – Relações entre entidades de domínio ambiente, domínio e perfis	39
Figura 4 – Entidades de domínio: tipos de perfis	40
Figura 5 – Entidades de domínio: tipos de artigos	41
Figura 6 – Estrutura de Plugins	41
Figura 7 – Exemplo do uso do Comunidade.UnB na disciplina de MES.	43
Figura 8 – Resultados do questionário para a pergunta 1	54
Figura 9 – Resultados do questionário para a pergunta 2	54
Figura 10 – Resultados do questionário para a pergunta 3	55
Figura 11 – Resultados do questionário para a pergunta 4	55
Figura 12 – Issue 121: Gerenciamento de tempo para <code>work_assignment</code>	65
Figura 13 – Merge Request: Gerenciamento de tempo para <code>work_assignment</code>	66
Figura 14 – Issue 119: Adicionar notas ao <i>plugin Work Assignment</i>	67
Figura 15 – Merge Request: Sistema de notas e módulos para <code>work_assignment</code>	68
Figura 16 – Tela para criação de grupos de trabalhos a serem enviados.	69
Figura 17 – Tela principal de um módulo.	70
Figura 18 – Lista de atividades de um <i>work assignment</i>	70
Figura 19 – Tela para criação de um <i>work assignment</i>	71
Figura 20 – Representação do estado dos trabalhos a serem entregues.	71
Figura 21 – Tela para o gerenciamento de avaliações.	72
Figura 22 – Tela para atribuição e alteração da notas.	73
Figura 23 – Visualização das notas de trabalhos enviados.	73
Figura 24 – Tela para visualização das atividades de forma centralizada.	74
Figura 25 – Blocos para visualização das atividades.	74
Figura 26 – Diagrama de classes do <i>Plugin Work Assignment</i>	75
Figura 27 – Tela para associar matricula ao usuário.	77

Lista de tabelas

Tabela 1	–	Tabela de comparação categorias: Conteúdo e Atribuição	48
Tabela 2	–	Tabela de comparação categoria Curso	49
Tabela 3	–	Tabela de comparação categoria Ferramentas	50
Tabela 4	–	Tabela de comparação categoria Teste/Quiz	51
Tabela 5	–	Tabela de comparação categorias: Permissões e papéis, página principal	52
Tabela 6	–	Opções para preenchimento do questionário.	89

Lista de abreviaturas e siglas

AGPL	Affero General Public License
AVA	Ambiente Virtual de Aprendizagem
BDD	Behavior Driven Development
CPD	Centro de Informática
DRY	Don't Repeat Yourself
FGA	Faculdade UnB Gama
VLE	Virtual Learning Environment
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
IRC	Internet Relay Chat
Moodle	Modular Object-Oriented Dynamic Learning Environment
LAPPIS	Laboratório de Produção, Pesquisa e Inovação em Software
LDAP	Lightweight Directory Access Protocol
MVC	Model-View-Controller
NIED	Núcleo de Informática Aplicada à Educação
ORM	Object-relational mapping
Rails	Ruby on Rails
RAM	Random Access Memory
SGBD	Sistema Gerenciador de Banco de Dados
SMT	Tecnologias de Mídia Social
TDD	Test Driven Development
US	User Story
VLE	Virtual Learning Environment

Sumário

1	INTRODUÇÃO	21
1.1	Justificativa	21
1.2	Objetivos	22
1.2.1	Objetivo Geral	22
1.2.2	Objetivos Específicos	22
1.3	Organização do Trabalho	22
2	MANUTENÇÃO E EVOLUÇÃO DE SOFTWARE	25
2.1	Manutenção e evolução de software nos métodos tradicionais	25
2.2	Manutenção e evolução de software em métodos empíricos de desenvolvimento	26
2.2.1	Software Livre	27
2.2.2	Relações entre métodos ágeis e processo de desenvolvimento de Software Livre	28
2.2.2.1	Práticas baseadas em disciplinas da engenharia de software	30
3	AMBIENTE VIRTUAL DE APRENDIZAGEM E PLATAFORMAS DE REDES SOCIAIS	31
3.1	Ambiente virtual de aprendizagem	31
3.2	Redes sociais	32
4	EVOLUINDO UMA PLATAFORMA DE REDE SOCIAL	35
4.1	Noosfero	35
4.1.1	Práticas de desenvolvimento da comunidade Noosfero	35
4.1.2	Arquitetura	37
4.1.3	Modelo de domínio	39
4.1.4	Plugins	41
4.2	Comunidade UnB	42
5	ESTUDO PRELIMINAR	45
5.1	Comparação entre Noosfero e ambientes virtuais de aprendizagem	45
5.2	Pesquisa com alunos	53
5.3	Histórias de usuário	56
6	DESENVOLVIMENTO DE UMA REDE SOCIAL DE COLABORAÇÕES E APRENDIZAGEM	65
6.1	O processo de desenvolvimento	65
6.2	A evolução do <i>plugin Work Assignment</i>	69
6.2.1	Arquitetura do Plugin Work Assignment	75
6.3	A evolução do <i>plugin Comunidade.UnB</i>	76

7	CONCLUSÃO	79
7.1	Limitações	79
7.2	Trabalhos Futuros	80

	Referências	83
--	--------------------	-----------

	APÊNDICES	87
--	------------------	-----------

	APÊNDICE A – INSTRUÇÕES PARA PREENCHIMENTO DO QUESTIONÁRIO	89
--	---	-----------

	APÊNDICE B – QUESTIONÁRIO PARA PROBLEMATIZAÇÃO	91
--	---	-----------

1 Introdução

Nas últimas décadas houve ampliação dos recursos computacionais e do acesso à *Internet*, por conseguinte houve um aumento significativo na quantidade e diversidade de conteúdos e serviços à disposição dos usuários. Araújo (2011) menciona que um dos fatores para esse aumento é que indivíduos que anteriormente limitavam-se ao papel de consumidores de conteúdo, hoje colocam-se também na posição de produtores. Neste contexto, encontramos na Internet inúmeros exemplos de sucesso como *blogs*, enciclopédias colaborativas como a Wikipedia, repositórios para compartilhamento de fotografia e vídeo, a exemplo Flickr e Youtube e redes sociais colaborativas.

Castells (2007) afirma que tal fenômeno, comumente referenciado como *Web 2.0*, ocorre porque a população acredita que pode influenciar outras pessoas, atuando no mundo por meio da sua força de vontade e utilizando seus próprios meios. Isso explica o crescimento no uso de redes sociais nos últimos anos, onde as pessoas querem produzir seus próprios conteúdos, gerando discussões construtivas sobre diversos assuntos.

O ambiente acadêmico atual faz uso de ferramentas virtuais de aprendizagem, soluções colaborativas e redes sociais para interação. Essas ferramentas auxiliam professores e alunos em suas atividades diárias, facilitam a rápida atualização de conteúdos, facilitam o acesso a públicos geograficamente dispersos, reduz custos logísticos e administrativos e auxiliam aos alunos no desenvolvimento de capacidades de auto estudo e autoaprendizagem. Nesse contexto, deve-se buscar solução integrada, de fácil acesso e que agregue as diversas funcionalidades necessárias a um ambiente virtual de aprendizagem.

1.1 Justificativa

A Universidade de Brasília (UnB) faz uso de ambiente virtual de aprendizagem, suportado pela ferramenta moodle, que permite a criação de cursos “on-line”, páginas de disciplinas, grupos de trabalho e aprendizagem virtual. Essa ferramenta é um suporte tecnológico utilizada pelos professores como suporte a condução de suas disciplinas. Baseado em métodos empíricos, verifica-se que nesse contexto os alunos a utilizam, apenas para consumirem conteúdos produzidos ou responderem a estímulos oriundos dos professores.

Foi aplicado um questionário para avaliar se os alunos da Universidade de Brasília utilizam redes sociais, como o *Facebook*, para o compartilhamento de recursos e informações relacionados as disciplinas cursadas na Universidade. Constatou-se que 83% dos alunos utiliza as redes sociais para tal fim, mesmo não havendo recomendação dos professores. (Resultados no Capítulo 5). O resultado do questionário vai ao encontro da

afirmação de [Davis et al. \(2012\)](#), de que os alunos de graduação querem estar nas redes sociais e que aqueles que participam de forma mais ativa nessas redes tendem a obter maior riqueza em suas relações sociais.

Para atender este novo contexto onde os usuários também querem produzir seus conteúdos e trabalharem de maneira horizontal e colaborativa, a UnB também possui uma rede colaboração livre Comunidade.UnB , desenvolvida para que alunos, professores e servidores técnico-administrativos tenham um ambiente virtual de criação e compartilhamento de conhecimento colaborativo. Essa comunidade foi implementada com base na plataforma Noosfero, que permite que o usuário tenha o seu espaço, com total flexibilidade de personalização visual e gerenciamento de conteúdo.

Neste contexto, este projeto adiciona funcionalidades para que os alunos permaneçam nas redes sociais, mas no âmbito de assuntos acadêmicos, por meio da adição de funcionalidades no Comunidade.Unb, relativas a ambientes virtuais de aprendizagem. Para propor um hibridismo entre uma rede social colaborativa da UnB e um ambiente virtual de aprendizagem, dessa maneira há necessidade de evolução da plataforma Noosfero, que não dispõe de alguns recursos necessários para esses ambientes.

1.2 Objetivos

1.2.1 Objetivo Geral

Evoluir o Noosfero para permitir que ele contenha funcionalidades de um Ambiente Virtual de Aprendizagem (AVA).

1.2.2 Objetivos Específicos

- Identificar as funcionalidades presentes em AVA.
- Comparar as funcionalidades dos AVA com as disponíveis no Noosfero.
- Selecionar as funcionalidades dos AVA a serem implementadas no Noosfero.
- Implementar as funcionalidades selecionadas para os usuários da Comunidade.UnB.

1.3 Organização do Trabalho

Nesta seção apresenta-se como este trabalho está organizado e o que será abordado em cada um dos capítulos. O trabalho se inicia com o Capítulo 2, no qual são abordados conceitos relacionados à manutenção e evolução de software nos métodos

tradicionais e empíricos de *software* além de descrições de práticas que foram utilizados no desenvolvimento deste projeto. No capítulo 3 são discutidos os conceitos de redes sociais e AVA, além do levantamento de suas principais funcionalidades. No Capítulo 4 será apresentado a arquitetura e funcionamento da plataforma Noosfero, utilizado pela Comunidade.Unb. No Capítulo 5 é apresentado o estudo preliminar realizado para o desenvolvimento incluindo uma comparação de funcionalidades entre AVA e o Noosfero, afim de verificar quais delas a plataforma carece. Além das propostas de melhoria e evolução deste trabalho para com a plataforma. No Capítulo 6 são apresentados o processo realizado durante o desenvolvimento e os resultados obtidos na evolução dos *plugins*. O trabalho se encerra no Capítulo 7 com a conclusão, assim como as limitações encontradas, além de uma discussão e trabalhos futuros, que podem incrementar esse trabalho.

2 Manutenção e evolução de software

Neste capítulo será abordada a manutenção e evolução de software, fornecendo uma comparação entre os métodos tradicionais e empíricos empregados na engenharia de software. Também serão evidenciadas práticas adotadas nas comunidades de *software* livre, que serão utilizadas neste trabalho.

2.1 Manutenção e evolução de software nos métodos tradicionais

A [ISO/IEC \(1998\)](#) 12207 define o ciclo de vida de um software por meio de agrupamento de processos que são divididos em quatro classes: fundamentais, apoio, organizacionais e adaptação. O processo de manutenção de software está classificado nessa ISO como um dos processos fundamentais, ou seja, necessários para que um o software seja implementado.

O Instituto de Engenheiros Eletricistas e Eletrônicos [IEEE 1219 \(1998\)](#), define manutenção de software como a modificação de um produto de software após sua entrega com vistas a corrigir falhas, melhorar o desempenho ou outros atributos, e adaptar o produto a um ambiente modificado de acordo com as necessidades do cliente.

As normas citadas tratam a manutenção de software como se o mesmo fosse um produto genérico. Para [Pfleeger \(2004, p. 380\)](#) a manutenção varia de acordo com o propósito de criação do software. Ainda, segundo a Autora, a probabilidade de o sistema ser modificado varia de acordo com o vínculo de dependência dos requisitos com o sistema do mundo real, pois, quanto mais dependentes forem os requisitos do mundo real maior será a probabilidade do sistema ser modificado.

Para determinar o tipo de manutenção a ser realizado, [Pfleeger \(2004\)](#) separa sistemas em três grupos. O primeiro grupo de sistemas é o que possui os requisitos especificados e formalmente definidos, ou seja, os resultados esperados são bem conhecidos e a solução é estática e não se adapta facilmente a uma mudança no problema que o gerou. O segundo grupo consiste dos sistemas, cujas soluções são aproximadas do mundo real, tem-se a solução completa apenas na teoria pois solucionar o problema em sua totalidade é impraticável ou impossível. O jogo de xadrez é um exemplo desse grupo, uma vez que é impossível calcular todas as movimentações possíveis e suas consequências em tempo hábil para a próxima jogada. O terceiro grupo incorpora a mudança na natureza do mundo real em si, modificando-se a medida em que o mundo muda. Tem como base um modelo do processo abstrato envolvido. Exemplo: sistema que prevê a estabilidade econômica de um país e utiliza como base o modelo econômico em uso.

Analisando a literatura percebe-se que a manutenção de software está dividida em três grandes categorias, listadas a seguir, que podem ser visualizados tanto na norma da [IEEE 1219 \(1998\)](#) e em livros, desde [Lientz e Swanson \(1980\)](#) à [Pfleeger \(2004\)](#), as categorias são as seguintes:

- **Manutenção corretiva:** realizada para controlar funções cotidianos do sistema, corrigindo falhas e defeitos de funcionalidades do programa.
- **Manutenção adaptativa:** visa a implementação de modificações secundárias que têm como origem uma modificação em outra parte do sistema ou do meio externo.
- **Manutenção perfectiva ou evolutiva:** objetivam realizar acréscimo de novas funcionalidades ou mudanças no *software* independente da existência de defeitos.

[Pressman \(2011\)](#) e [Criscuolo \(2008\)](#) acrescentam a manutenção preventiva. Essa está ligada a uma reengenharia, aplicando conceitos de engenharia de software para tornar os softwares mais fáceis de serem corrigidos, adaptados e melhorados ao longo dos anos. Para [Criscuolo \(2008\)](#) a manutenção preventiva permite que o software receba alterações com objetivo de minimizar os riscos de defeitos e reduzir a complexidade de sua estrutura, garantindo uma certa segurança à manutenção.

Para o [IEEE 1219 \(1998\)](#) além das categorias citadas, há a manutenção “emergencial”, definida pelo IEEE como uma manutenção corretiva que não foi programada ou planejada, mas que deve ser executada para manter o sistema em funcionamento.

As atividades do processo de desenvolvimento de software ainda são dependentes da participação e criatividade humana, e o crescimento de métodos ágeis e software livre mostram que o código é um dos principais artefatos. Diante disso torna-se necessário na Engenharia de *Software* utilizar métodos empíricos que apoiam o desenvolvimento de *software* de acordo com sua natureza abstrata.

2.2 Manutenção e evolução de software em métodos empíricos de desenvolvimento

A ciência empírica diz respeito à aquisição de conhecimentos através de métodos empíricos. No entanto, o que constitui o conhecimento, assim sendo, os métodos para adquiri-lo baseia-se em pressupostos básicos sobre ontologia (ou seja, o que acreditamos existir) e epistemologia (ou seja, como as crenças são adquiridas e que justifica-los). Dessa maneira a pesquisa empírica procura explorar, descrever, prever e explicar fenômenos naturais, sociais, cognitivas ou usando evidências baseadas em observação ou experiência. Trata-se de obtenção e interpretação de evidências, por exemplo: experimentação; obser-

vação sistemática; entrevistas ou inquéritos; ou pela análise minuciosa de documentos ou artefatos ([SJOBERG](#); [DYBA](#); [JORGENSEN](#), 2007).

Neste capítulo serão abordadas algumas metodologias e práticas no processo de desenvolvimento de software que advém de métodos empíricos, que neste contexto destaca-se os métodos ágeis e o processo de desenvolvimento de software livre.

2.2.1 Software Livre

De acordo com [Meirelles \(2013\)](#) software expressa uma solução abstrata dos problemas computacionais, neste contexto é o componente que contém o conhecimento relacionado aos problemas a que a computação se aplica, contendo diversos aspectos que ultrapassam questões técnicas, tais como:

- o processo de desenvolvimento do software;
- os mecanismos econômicos que regem o desenvolvimento e o uso do software, a exemplo de gerenciais, competitivos, sociais, cognitivos etc.;
- o relacionamento entre desenvolvedores, fornecedores e usuários do software;
- os aspectos éticos e legais relacionados ao software.

O que define e diferencia o software livre de software proprietário vai do entendimento desses quatro pontos dentro do que é conhecido como *ecossistema do software livre* ([MEIRELLES](#), 2013).

Software livre é definido pela Free Software Foundation como “uma questão de liberdade dos usuários para executar, copiar, distribuir, estudar, mudar e melhorar o software. Levando isto em consideração significa que os usuários do software possuem quatro liberdades essenciais ([STALLMAN](#), 2002):

Liberdade no. 0 Executar o programa, para qualquer propósito;

liberdade no. 1 Estudar como o programa funciona, e adaptá-lo para as suas necessidades;

liberdade no. 2 Redistribuir cópias de para ajudar o próximo;

liberdade no. 3 Aperfeiçoar o programa, e liberar os seus aperfeiçoamentos, de modo que toda a comunidade se beneficie. Acesso ao código-fonte é um pré-requisito para esta liberdade.

Para que as liberdades um e três façam sentido, é necessário que o desenvolvedor ou pesquisador tenha acesso ao código-fonte do programa. Portanto, acesso ao código-fonte é uma condição necessária ao software livre (GNU, 2013).

Um programa é considerado um software livre se os usuários possuem todas as liberdades citadas. Dessa forma, você deve ser livre para redistribuir cópias, com ou sem modificações, de graça ou cobrando uma taxa pela distribuição, para qualquer lugar. Ser livre para fazer essas coisas significa, entre outras coisas, que não há necessidade de pedir ou pagar por permissão para fazê-las (SANTOS, 2012).

O fato de que o código-fonte pode ser livremente compartilhado oferece vantagens ao software livre em comparação ao software proprietário. Uma delas é simplificação do desenvolvimento de aplicações personalizadas já que não precisam ser programadas a partir do zero, ou seja, podem se basear em soluções existentes.

A outra vantagem resultante do compartilhamento do código refere-se à possibilidade de melhoria na qualidade, em particular frente aos problemas inerentes à sua complexidade (RAYMOND, 1999).

Isso se deve ao maior número de desenvolvedores e usuários envolvidos com o software, permitindo maiores situações de uso em necessidades variadas, o que propicia a identificação de um número maior de *bugs* e mais sugestões de melhoria, promovendo refatorações que geralmente levam a melhoria do código.

O projeto GNU (é um acrônimo recursivo de: GNU is Not Unix) teve como objetivo criar um sistema operacional, chamado GNU, baseado em software livre. Para isso veio a necessidade de utilizar termos de distribuição que impedissem a transformação de software livre em proprietário. O método usado foi chamado de *Copyleft*. *Copyleft* é um método geral para desenvolver um software livre e exige que todas as versões modificadas e estendidas do programa também sejam livres. Ele utiliza lei de direitos autorais, com finalidade diferente da usual: ao invés de um meio de privatizar o software, torna-se um meio de manter o software livre (STALLMAN; GAY, 2009).

2.2.2 Relações entre métodos ágeis e processo de desenvolvimento de Software Livre

Os métodos ágeis assim como processo de desenvolvimento de software livre, são métodos empíricos de software que provêm de um conjunto de metodologias baseada na prática de desenvolvimento de software. Em 2001 um grupo de desenvolvedores formaram a aliança ágil e definiram o Manifesto ágil, que definem preferências e princípios que se espera de qualquer método de desenvolvimento dessa categoria. O manifesto é baseado em quatro valores (BECK et al., 2001):

- Indivíduos e interações são mais importantes que processos e ferramentas.
- Software em funcionamento é mais importante que documentação abrangente.
- Colaboração com o cliente (usuários) é mais importante que negociação de contratos.
- Responder às mudanças é mais importante que seguir um plano.

Na dissertação de [Corbucci \(2011\)](#) há evidências de que esses quatro valores demonstram a semelhança entre as práticas comuns utilizadas pelas comunidades de software livre e equipes ágeis. Além disso, o Autor, destaca que muitas práticas disseminadas pelas metodologias ágeis são usadas no dia a dia dos desenvolvedores e equipes das comunidades de software livre:

- código compartilhado (coletivo);
- projeto simples;
- repositório único de código;
- integração contínua;
- código e teste;
- desenvolvimento dirigido por testes;
- refatoração.

Entender esses aspectos no contexto de software livre torna-se relevante porque, ao contrário dos métodos tradicionais de desenvolvimento, o ágeis são adaptativos ao invés de prescritivos e são orientados às pessoas ao invés dos processos. Analisar o processo de desenvolvimento de software livre do ponto de vista da Engenharia de Software e as possíveis sinergias com os métodos ágeis podem contribuir para um melhor entendimento dessa disposição na criação e colaboração em torno de projetos de software livre ([MEIRELLES, 2013](#)).

No processo de desenvolvimento de software livre, após a divulgação e lançamento da primeira versão do projeto, a solução está pronta para uso. Os usuários são desenvolvedores podem colaborar com o projeto afim de evoluí-lo para suprir necessidades pessoais ou de interesse da comunidade, são enviadas aos mantenedores do projeto como *patches*¹. Os mantenedores analisarão as propostas, e caso concordem com a mudança e implementação, irão aplicá-las ao repositório oficial do projeto. Portanto, mesmo que em projetos

¹ arquivos que contém as modificações no código

maiores outros aspectos sejam levados em consideração ou sigam processos mais burocráticos de colaboração, a essência da colaboração técnica está no envio e análise de trechos de código-fonte (MEIRELLES, 2013).

As comunidades de software livre são “livres” para escolher e adotar as práticas que mais se adaptam ao seu modelo, dessa maneira é relatado o processo de desenvolvimento do Noosfero, um exemplo de desenvolvimento de software livre evidenciado na seção 4.1.1 do capítulo 4.

2.2.2.1 Práticas baseadas em disciplinas da engenharia de software

No contexto de gerência e configuração de software grande parte dos projetos de software livre atualmente utilizam algum sistema de controle de versão, visto como uma extensão natural do processo de desenvolvimento. Esse sistema, normalmente faz-se uso de um repositório central, onde são armazenadas todas as versões dos arquivos, e permite paralelizar o desenvolvimento de forma organizada, principalmente quando há um grande número de desenvolvedores (REIS; FORTES, 2001). Permite ainda o *commiter* oficial realizar inspeção e revisão de todo código que é integrado ao repositório principal do projeto de forma não intrusiva no desenvolvimento realizado pelos colaboradores. Reis e Fortes (2001) atestam que a revisão acaba acontecendo quase automaticamente, já que as pessoas são forçadas pela distância a enviar modificações a outras para a integração.

O teste de software é um processo destinado a certificar-se de que o código desenvolvido é previsível e consistente, e que realiza apenas as ações para o qual foi projetado (MYERS; SANDLER; BADGETT, 2011). Teste automatizado é uma prática de criar *scripts* ou programas de computador que exercitam o sistema em teste, capturam os efeitos colaterais e fazem verificações, de forma automática e dinâmica (MESZAROS, 2007).

Segundo Bernardo (2011) os testes automatizados agregam valor ao produto final porque afetam diretamente a qualidade dos sistemas de software, mesmo que os artefatos produzidos não sejam visíveis aos usuários.

A integração contínua tem por objetivo integrar o sistema e todas suas dependências para assegurar que nenhuma modificação tenha danificado o sistema, sejam elas alterações no código-fonte, em configurações ou mesmo em dependências e outros fatores externos (DUVALL; MATYAS; GLOVER, 2007). Desse modo, para certificar a integridade do código a integração contínua garante que todos os testes são executados a cada modificação da base de código fornecendo *Feedback* imediato, assegurando que a integração tenha ocorrido de forma satisfatória.

Corbucci (2011) afirma que a equipe utiliza os testes para descobrir eventuais defeitos o mais rapidamente possível, o que facilita e acelera a correção, além de reduzir a

probabilidade de que pequenos problemas se transformarem em *bugs* ou comportamentos inesperados no futuro.

Baseado nas disciplinas de gerência e configuração e teste, fazendo uso de controle de versão, testes automatizados e integração contínua, é que evoluíremos a plataforma Noosfero com as funcionalidades selecionadas para melhor adaptar a plataforma com recursos de AVA. Para isso será abordado no próximo capítulo (3) os principais conceitos e recursos dos AVA.

3 Ambiente Virtual de Aprendizagem e Plataformas de Redes Sociais

Neste capítulo serão apresentadas os conceitos relacionados aos ambientes virtuais de aprendizagem e redes sociais. Além disso há uma comparação dos principais AVA levantados nessa pesquisa com a plataforma Noosfero.

3.1 Ambiente virtual de aprendizagem

No atual cenário, em que vivemos a tecnologia da informação está cada vez mais presente em nossa rotina, seja com o uso de celulares, tablets, televisores entre outros e utilizá-la no ambiente educacional pode ser uma grande vantagem de integração à sociedade do conhecimento. Nesse cenário ambientes virtuais de aprendizagem tem sido cada vez mais usados nas mais diversas instituições (acadêmicas, empresariais e tecnológicas) com o objetivo de apoiar o ensino à distância.

Os ambientes virtuais de aprendizagem ¹ são sistemas que tem a funcionalidade de *software* para a comunicação entre computador e métodos de entrega de material de cursos online. Muitos desses ambientes permitem simular a sala de aula física em um meio virtual contudo existem ambientes que além de realizar este papel fazem uso da tecnologia para propiciar aos envolvidos novas ferramentas que potencializem a comunicação e facilitem a aprendizagem (SCHLEMMER, 2005).

Os AVA são utilizadas por várias instituições de ensino do mundo. O mercado apresenta uma grande variedade de *softwares* proprietários e livres. Entretanto, apesar dessa diversidade, Behar e BASSANI (2009) destaca que muitas características e ferramentas são comuns entre eles:

- permitem acesso restrito a usuários previamente cadastrados;
- disponibilizam espaço para a publicação de material do professor;
- disponibilizam espaço destinado ao envio/armazenamento de tarefas realizadas pelos alunos;
- disponibilizam um conjunto de ferramentas de comunicação síncrona e assíncrona²;
- disponibilizam correio eletrônico (*e-mail*) e mural de recados.

¹ Em inglês *Virtual Learning Environment* (VLE)

² Como *chat* e fórum de discussões

Além dessas características, [Aguado \(2013\)](#) ressalta que os principais componentes encontrados consistem de:

- disponibilização de conteúdo de ensino;
- mapeamento do conteúdo do curso (quebra do conteúdo em sessões que podem ser acessadas e cumpridas);
- disponibilização de atividades de interação e avaliação;
- acompanhamento do desempenho do aluno;
- suporte online;
- links de internet para conteúdos externos;
- identificações e permissões de acesso.

Apesar dos autores supracitados relatarem as principais características e componentes dos AVA, constata-se que nem todos oferecem as mesmas funcionalidades como pode ser verificado na seção 5.1 desse trabalho. O escopo das opções disponíveis nos sistemas nem sempre é o mesmo, evidenciando a razão do alto número de sistemas no mercado, pois os produtos possuem pontos fortes e fracos, e por isso, nem sempre as funcionalidades atendem aos diversos tipos de instituições e seus respectivos objetivos ([AGUADO, 2013](#)).

Segundo [Chin \(2003\)](#), as funcionalidades oferecidas pelo AVA facilitam e fortalecem o aprendizado de conceitos dados presencialmente. Mesmo que os AVA apresentem características principais do ensino tradicional em sala de aula, [Chin \(2003\)](#) propõe que os AVA não devem ser vistos como substitutos à aula presencial, uma vez que sempre deverá existir a interação humana física no processo educacional.

Dentre os vários sistemas disponíveis no mercado alguns necessitam de instalação no computador cliente, porém, a maioria é acessível pela internet, onde a instalação é realizada apenas no servidor. Das ferramentas encontradas atualmente destacam-se as tradicionalmente utilizadas como o Moodle ³ e o BlackBoard ⁴ detalhadas na seção 5.1.

3.2 Redes sociais

Para entender o conceito de redes sociais é importante destacar que são apenas uma camada das mídias sociais. É comum vermos a utilização deste termo para todos os tipos de mídias sociais mediadas por um computador. As tecnologias de mídias sociais ⁵

³ *Modular Object-Oriented Dynamic Learning Environment*. Disponível em: <<https://moodle.org/>>

⁴ Disponível em: <<http://www.blackboard.com/>>

⁵ Termo em inglês, *Social Media Technology* (SMT)

podem ser definidas de forma abrangente como a totalidade de produtos e serviços digitais disponibilizados online, como aplicações *web* e *mobile* que permitem aos indivíduos criar e compartilhar conteúdos onde sua principal fonte são os usuários, através da comunicação em várias vias (DAVIS et al., 2012).

As definições de rede utilizadas neste trabalho estão relacionadas com três propostas. A primeira é de acordo com Emirbayer e Goodwin (1994) que definem rede como um conjunto de relações ou ligações entre um conjunto de atores, no caso atores são os elos às pessoas que se comunicam em uma dada rede. No mesmo sentido, Costa (2004, p. 73) a define com uma forma de organização caracterizada fundamentalmente pela sua horizontalidade, isto é, pelo modo de inter-relacionar os elementos sem hierarquia. E por fim, mas não menos importante, faz-se uso da definição proposta por Tomaél, Alcará e Chiara (2005) no qual rede é uma estrutura não-linear, descentralizada, flexível, dinâmica, sem limites definidos e autoorganizáveis. Destacando que uma rede estabelece-se por relações horizontais de cooperação.

Segundo (MARTELETO, 2001), as redes sociais derivam dos conceitos de rede e representam um conjunto de participantes autônomos, que buscam unir suas idéias e recursos em torno de valores e interesses compartilhados. A autora ressalta a ideia de compartilhamento de valores e interesses que, para promover o fortalecimento da rede, dependem do compartilhamento da informação e do conhecimento.

Nesse contexto, os autores Ellison et al. (2007) definem sites de redes sociais como serviços *web* que permitem a seus usuários criarem perfis e, por meio deles, conectarem com outros usuários, propiciando a busca e o cruzamento de informações dentro da sua lista de conexões.

Fundamentado nestes principais conceitos, este trabalho está focado em sites de redes sociais, que possuem o objetivo de promover a interação em torno das colaborações, isto é, apoiada no entendimento de redes de colaboração. Isto se aplica ao contexto de uma rede social de nicho de uma universidade, como explicitado na seção 4.2 onde é descrita a proposta de disponibilização de uma rede social para a Universidade de Brasília (UnB). A rede da UnB permite que as pessoas entrem na rede para acompanhar uma disciplina, um projeto ou um determinado grupo de trabalho. Os alunos também podem explorar a rede para encontrar comunidades e conteúdos de seus interesses.

Bucher (2013) menciona que esta prática adotada por redes de colaboração não é possível em redes monopolistas e centralizadoras, porque em geral seus conteúdos estão dipostos de forma fragmentada ou são controlados. Isto demonstra que o uso de uma rede de colaboração dentro de universidades é uma escolha mais plausível para a difusão do conhecimento.

Apoiado nas vantagens proporcionadas pelas redes sociais de colaboração e sua

implantação na UnB, busca-se neste trabalho associá-las aos conceitos dos ambientes virtuais de aprendizagem tratados na Seção 3.1. A proposta visa a criação de funcionalidades que viabilizem o uso de uma plataforma de redes sociais como um ambiente de apoio aos ambientes virtuais de aprendizagem.

Neste capítulo foi apresentado os principais conceitos de ambientes virtuais de aprendizagem e redes sociais, no entanto é necessário explorar o Noosfero (Capítulo 4) assim como os seus processos e desenvolvimento dentro da comunidade de *software* livre. Esse conhecimento é essencial, uma vez que utilizaremos o Comunidade.UnB, que utiliza essa plataforma sendo o foco principal desse trabalho.

4 Evoluindo uma plataforma de rede social

Neste capítulo é discutida a plataforma Noosfero, desde sua arquitetura até os processos de desenvolvimento da comunidade. Além disso, é apresentado o Comunidade.UnB, rede de colaboração livre, no qual são propostas evoluções para que se torne um ambiente híbrido utilizado por professores e alunos.

4.1 Noosfero

O Noosfero ¹ é uma plataforma livre para criação de redes sociais, desenvolvida em 2007, pela Cooperativa de Tecnologias Livres - Colivre ², sob a licença AGPL³ V3, com a proposta de facilitar a criação de redes sociais personalizadas, livres e autônomas e a geração de conteúdo colaborativo.

Além das funcionalidades de rede social, com foco na produção e compartilhamento de conteúdo, o Noosfero permite que dentro da rede cada usuário e comunidade tenha o seu espaço com total flexibilidade de personalização visual e gerenciamento de conteúdo. São exemplos de portais que utilizam o Noosfero: o Participa BR ⁴, Stoa⁵, Portal da FGA⁶ e o novo Portal do Software público Brasileiro ⁷.

O Noosfero foi desenvolvido na linguagem *Ruby*, na versão 2.2.0, pois possui uma sintaxe simples, que facilita a manutenibilidade do sistema, característica importante em projetos de software livre que tendem a atrair colaboradores externos (MEIRELLES, 2013). Também utiliza o *framework* aplicações web *Ruby on Rails* ⁸, versão 4. E utiliza também padrões arquiteturais de software Model-View-Controller (MVC), e o padrão de plugins que serão apresentados na seção 4.1.2. Além disso, por questões de segurança, o Noosfero é homologado para a versão *stable* do *Debian*, acompanhando a versão do *Ruby* e do *Rails* dessa distro.

4.1.1 Práticas de desenvolvimento da comunidade Noosfero

Para que novos desenvolvedores colaborem com o Noosfero, a comunidade utiliza em seu próprio *site* uma seção para o desenvolvimento. Nessa seção, encontram-se os

¹ Disponível em: <<http://www.noosfero.org>>

² Disponível em: <<http://www.colivre.coop.br>>

³ Licença de software GNU

⁴ Disponível em: <<https://www.participa.br/>>

⁵ Disponível em: <<http://stoa.usp.br/>>

⁶ Disponível em: <<http://fga.unb.br/>>

⁷ Disponível em: <<https://portal.softwarepublico.gov.br/>>

⁸ Disponível em: <<http://rubyonrails.org/>>

manuals com os passos para instalação do ambiente, descrição dos *plugins* disponíveis, instruções para a personalização da plataforma através de temas além de informações arquiteturais da plataforma. Para realizar o controle de itens a fazer, como a implementação de novas funcionalidades ou correção de bugs é utilizado um *Issue Tracker* do repositório oficial no GitLab ⁹.

Uma vez que o desenvolvedor ou usuário tenha registro no GitLab, é possível utilizar o *Issue Tracker* para cadastrar novas funcionalidades ou *bugs* de maneira simples seguindo os seguintes passos:

1. preencher o campo título;
2. definir a descrição do item, se existir, é necessário especificar com qual *plugin* o item está relacionado;
3. associar um desenvolvedor responsável pela sua implementação *issue*;
4. definir as *labels* da funcionalidade, onde é especificado ao que está relacionado a nova *issue*.

Após a criação da *issue*, todos os membros da comunidade podem visualizar os dados e especificações, bem como discutir sobre os propósitos e formas de implementação da funcionalidade.

A Figura 1 evidencia o *Issue Tracker* do Noosfero onde é possível verificar os itens que foram mapeados e seus respectivos *status*, se ainda estão abertas ou fechadas, além de um filtro para os autores e os marcadores envolvidos à cada uma delas. Dessa forma, é possível priorizar os itens identificados como sendo de maior importância para os usuários e desenvolvedores.

Priorizando a comunicação entre os desenvolvedores, a comunidade compartilha canais ¹⁰ de comunicação pelo IRC (*Internet Relay Chat*). Nesses os desenvolvedores podem compartilhar conhecimento e realizarem discussões técnicas sobre as implementações.

A implementação é realizada pelo desenvolvedor que tem a responsabilidade de manter a qualidade do código produzido bem como realizar todos os testes relacionados à funcionalidade implementada. Uma vez que este primeiro passo esteja concluído o código é submetido a um *merge-request* onde um dos desenvolvedores do *core* efetua a revisão para verificar se está de acordo com os padrões esperados, e aprova ou não, a inclusão do código na *branch* principal do Noosfero.

⁹ Disponível em: <<https://gitlab.com/noosfero/noosfero>>

¹⁰ Canais do Noosfero: #noosfero-br e #noosfero

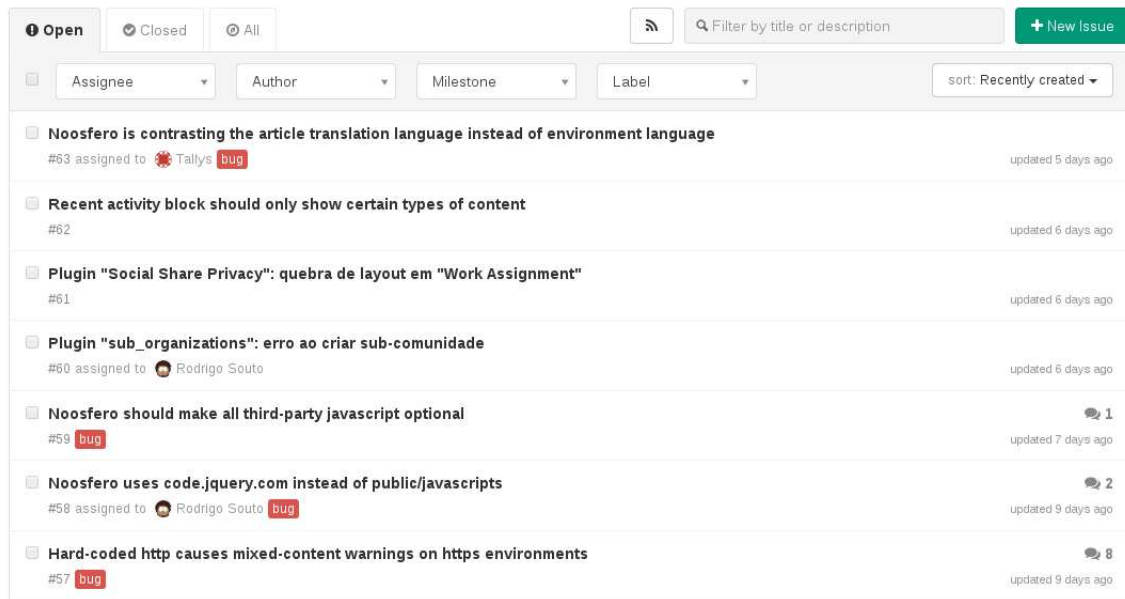


Figura 1 – Issue Tracker no GitLab

A comunidade Noosfero recomenda práticas de desenvolvimento como o TDD, *Test Driven Development* ou Desenvolvimento orientado a testes), combinado com o BDD ¹¹ (*Behavior Driven Development*) ou Desenvolvimento Guiado por Comportamento, que auxiliar o desenvolvedor a criar testes e integrar regras de negócio com a linguagem de programação, mantendo o foco no comportamento do software (NORTH, 2006).

Para realizar o controle de versão e gerenciamento do código fonte é utilizado o *Git*, uma ferramenta livre de versionamento distribuído de código fonte. O repositório oficial do Noosfero encontra-se no software livre Gitlab com um espelho no Github¹². Na página de desenvolvimento da comunidade existe uma série de recomendações sobre o envio de *patches* para o Noosfero, incluindo como versionamentos e solicitações de inclusão de seu *patch*, ou *merge-request*.

4.1.2 Arquitetura

Para a evolução de um software de forma adequada é importante o conhecimento da arquitetura do sistema, para não comprometer todo o planejamento realizado na concepção do projeto. Desse modo conhecer e entender a arquitetura de funcionamento do Noosfero é uma etapa fundamental para o desenvolvimento de novas funcionalidades para a plataforma.

A Figura 2 apresenta uma visão de alto nível da arquitetura do Noofero. Basicamente tem-se uma arquitetura cliente-servidor onde o cliente via *Browser* solicita um

¹¹ <<https://cukes.info/>>

¹² <<https://github.com/noosfero/noosfero>>

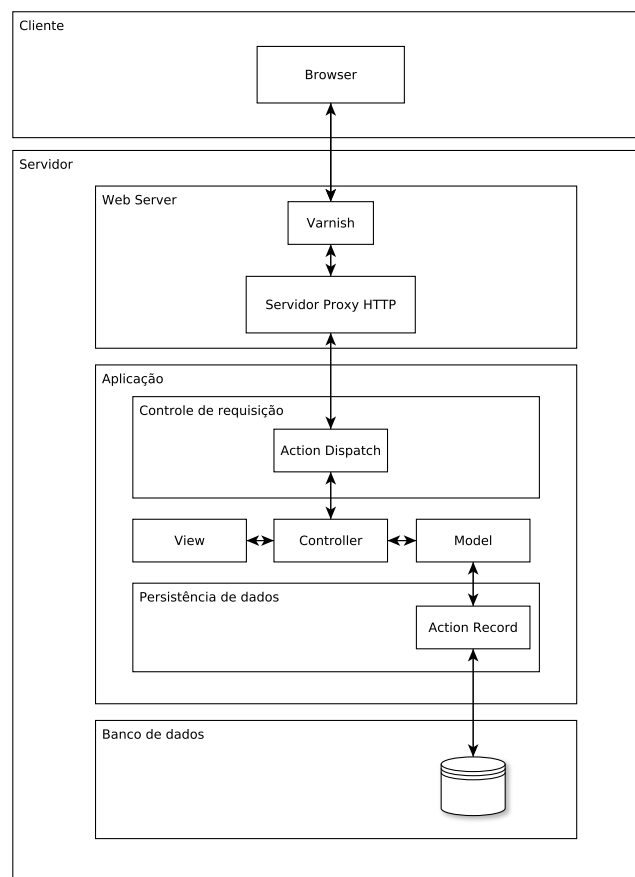


Figura 2 – Arquitetura do Noosfero

conteúdo ou uma função para o servidor Noosfero, que aguarda requisições de entrada para processá-las e compartilhar recursos com o cliente.

Do lado do servidor, temos basicamente três camadas de abstração: *Web-Server*, *aplicação* e *Banco de dados*. Na primeira camada temos dois componentes responsáveis por processar e acelerar todas as requisições de entrada e saída:

- Varnish: é um acelerador para sites web dinâmicos com alto volume de conteúdo que utiliza *proxy* HTTP Reverso. Sua eficiência deve-se ao fato dele armazenar o conteúdo HTTP requisitado na memória RAM, fazendo com que o servidor não consulte e processe diversas vezes o mesmo conteúdo solicitado.
- Servidor Proxy HTTP: pode ser utilizado o Apache ou Nginx ¹³ que ajudam a melhorar o desempenho funcionando como um servidor proxy HTTP reverso que processa as requisições de entrada e saída e as encaminha para a aplicação executá-las.

¹³ <<http://nginx.org>>

Na camada da aplicação, foi considerada uma camada responsável pelo controle de requisições, que é efetuada pelo componente **Action Dispatch**, que lida com o mapeamento de todas as requisições, *cookies* e sessão para suas respectivas *controllers*.

Na aplicação, utiliza-se o padrão de arquitetura de software MVC¹⁴ onde a **controller** controla o fluxo da aplicação, relacionando as entidades de *model* e de *view* através de chamadas de métodos. A **model** representa as entidades do domínio da aplicação, onde a lógica do sistema são implementadas. A **view** é a interface de comunicação com o usuário, ou seja as páginas HTML apresentadas no navegador.

Ainda na camada da aplicação tem-se o **Active Record** que é um ORM¹⁵, um mapeador entre objetos e registros de uma tabela, onde cada classe de modelo possui uma tabela correspondente à ela no banco de dados.

Por fim temos a camada de banco de dados que recebe requisições da camada de persistência de dados e por meio de um sistema gerenciador de banco de dados (SGBD) realizam operações na base de dados.

4.1.3 Modelo de domínio

Para Larman (2002, p. 160), um modelo de domínio é a representação visual de classes conceituais ou objetos do mundo real em um domínio, que também podem ser chamados de modelos conceituais, modelos de objetos de domínio e modelos de objetos de análise. Dessa forma, é necessário o seu entendimento para realizar a evolução da plataforma.

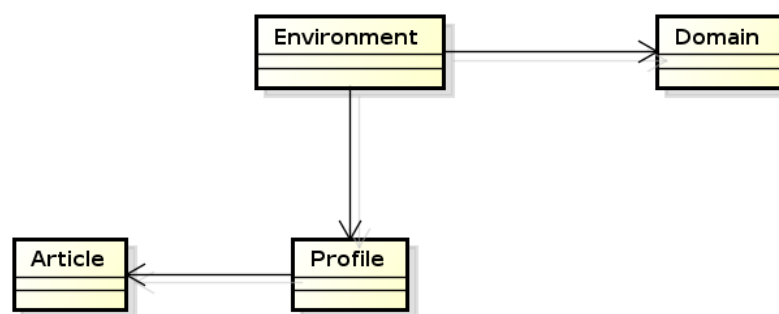


Figura 3 – Relações entre entidades de domínio ambiente, domínio e perfis. Extraído de: (BUCHER, 2013)

O Noosfero é uma plataforma que tem suporte a vários ambientes de rede social dentro de uma mesma instalação. A Figura 3 mostra-se o funcionamento geral do Noosfero com suas quatro principais classes **Domain**, **Environment**, **Profile** e **Article** (Em

¹⁴ Model-view-controller

¹⁵ object-relational mapping

português: Domínio, Ambiente, Perfil e Artigo respectivamente). Analisando o modelo verifica-se que na implementação é necessário que exista pelo um domínio e partir disso é possível criar várias instâncias de Ambiente na aplicação.

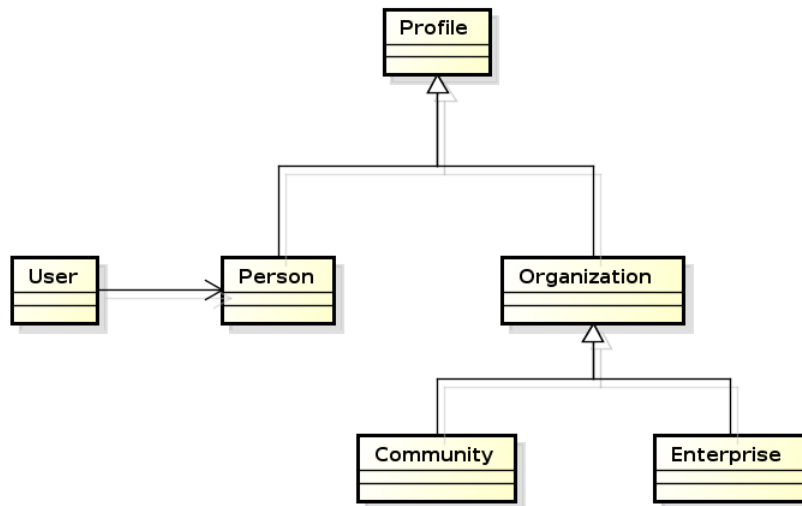


Figura 4 – Entidades de domínio: tipos de perfis. Extraído de: (BUCHER, 2013)

A entidade **Profile** é uma generalização das entidades **Person** (Pessoa) e **Organization** (Organização), como pode ser visto na Figura 4. Nesse mesmo modelo percebe-se que **Organization** é especializada nas entidades concretas **Community** (Comunidade e **Enterprise** (Empreendimento). A herança é um mecanismo pelo qual qual uma classe sub-classe pode estender uma super-classe, onde basicamente isola-se métodos ou atributos em comum dentro de uma classe pai (super-classe), enquanto as especialidades são responsabilidade das classes filhas (sub-classe).

Por questões de design do código da aplicação foi criada uma entidade **User**, ou Usuário, que é mantida separada da entidade Pessoa, que é quem implementa a lógica de autenticação da aplicação. Desta forma a lógica de autenticação fica separada da lógica de visualização e personalização do perfil.

Por fim, as entidades mostradas na Figura 5 representam os principais tipos de conteúdos disponíveis no Noosfero, onde a classe **Article**, ou Artigo, é uma especialização de todos os conteúdos disponíveis como: artigos de texto, pastas, blogs, galerias de imagens, fórum, arquivos e feeds de notícias.

O modelo de domínio aqui apresentado contempla o *core* do noosfero. Para o acréscimo de melhorias e funcionalidades é necessário compreender a visão arquitetural dos *plugins* d a plataforma, que será abordado na próxima seção.

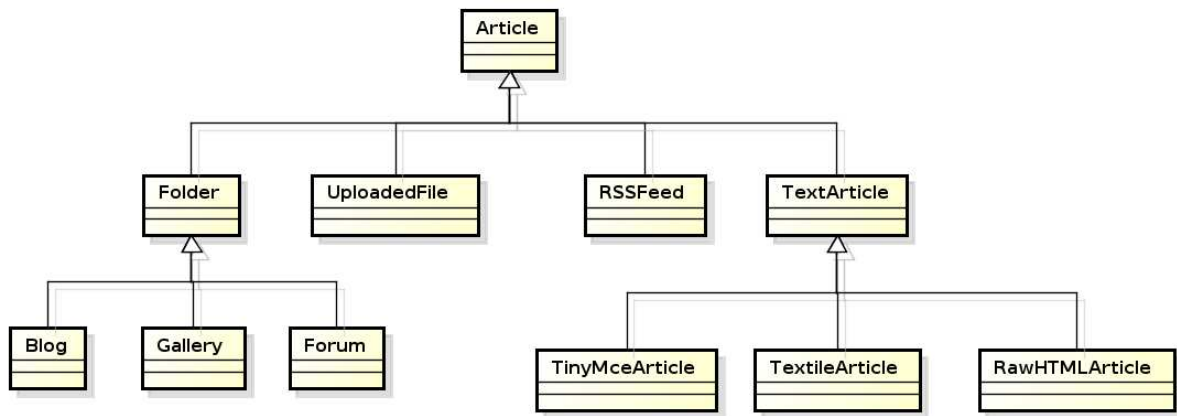


Figura 5 – Entidades de domínio: tipos de artigos. Extraído de: (BUCHER, 2013)

4.1.4 Plugins

Como é software em constante evolução, a arquitetura do Noosfero foi criada para ser altamente expansível, fazendo-se o uso de *plugins*. Essa arquitetura permite que em cada ambiente fique a critério do usuário quais os *plugins* ou novas funcionalidades serão habilitadas, o que torna o sistema flexível e modular.

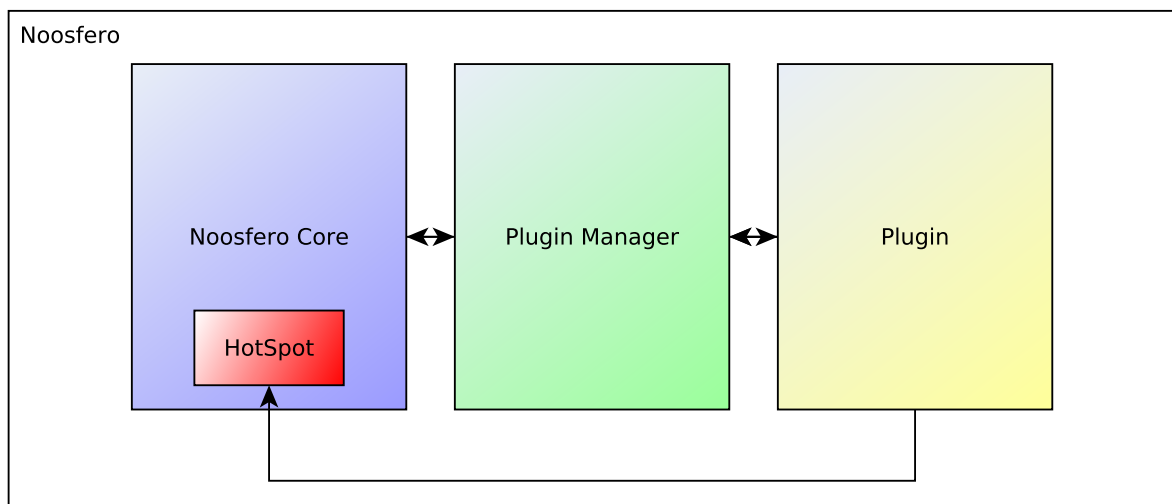


Figura 6 – Estrutura de Plugins

A Figura 6 é uma abstração alto nível do funcionamento interno dos plugins no Noosfero. No *core* do Noosfero temos os *hotspots*, que são pontos de flexibilidade que permitem associar diferentes comportamentos na execução do sistema, permitindo a inserção de trechos de código e ou alteração de um determinado método sem comprometer suas funcionalidades básicas.

Os *hotspots* são gerenciados por uma camada de abstração denominada *Plugin Manager*, ou Gerenciador de *Plugins*, que são chamadas pelo *core* através de um método

principal conhecido como *dispatch*. Basicamente, o ciclo de execução pode ser descrito da seguinte maneira: durante a execução de alguma funcionalidade o método *dispatch* é invocado por alguma funcionalidade do core, deste modo o gerenciador de plugins verifica todos os *Plugins* que fazem uso daquele *hotspot* e encaminha para cada um deles a execução de suas ações de acordo com sua implementação.

Essa arquitetura extensível adotada pelo Noosfero auxilia no controle da qualidade de código das novas funcionalidades. A camada de *Plugins* localiza-se fora do código do seu núcleo, em uma pasta denominada *plugins* em que desenvolvedor cria novas funcionalidades sem modificar o comportamento *core* do Noosfero, fazendo uso dos *dispatch*.

Adicionalmente como mencionado na Seção 4.1.1, o Noosfero faz uso de testes para manter a integridade de seu código, desse modo esta prática é estendida aos *plugins* que devem englobar seus respectivos testes para evitar a inserção de *bugs* e mudanças inesperadas no comportamento do sistema.

Assim sendo a evolução proposta para este trabalho será realizada através de plugins, como proposto na Seção 6

4.2 Comunidade UnB

A Comunidade.UnB é uma rede colaboração livre desenvolvida para que alunos, professores e servidores técnico-administrativos tenham um ambiente virtual de criação e compartilhamento de conhecimento colaborativo. É um ambiente virtual para o compartilhamento de ideias, produção de conteúdo colaborativo de modo que possam publicá-los para que possa ser de utilidade para outras pessoas ou parcelas da sociedade, uma vez que acredita-se que este é um dos papéis de uma Universidade (BUCHER, 2013).

Inspirada na rede social de colaboração Stoa¹⁶, da Universidade de São Paulo (USP), a Comunidade.UnB foi criada em 2013 a partir de um trabalho de conclusão de curso de Daniel Costa Bucher e até então está disponibilizada em ambiente de testes. Permite ao usuário a criação de seu espaço pessoal e a liberdade de publicar suas ideias, ou o conteúdo que desejar, por exemplo, na forma de blogs pessoais, blogs de disciplinas, pesquisas em andamento, dentre outras, além de compartilhar esse conteúdo para ser acessível para outros usuários dentro e fora da rede.

O Noosfero, descrito na Seção 4.1, foi a plataforma utilizada para o desenvolvimento da Comunidade.UnB, por dispor de um grande potencial e devido às suas funcionalidades avançadas, que permitem a criação e o compartilhamento de conteúdo de forma satisfatória. Além de dispor de uma comunidade ativa e de posição geográfica favorável em relação ao seu núcleo de desenvolvimento que se encontra no Brasil, facilitando

¹⁶ Disponível em: <<https://social.stoa.usp.br/>>

The screenshot displays the 'Comunidade UnB' website. At the top, there is a navigation bar with links: 'Entrar ou Registre-se', 'Pessoas', 'Comunidades', 'Conteúdos', and 'Eventos'. Below this, the 'Comunidade UnB' logo is on the left, and a search bar is on the right. The main content area features a large graphic of three interlocking shapes (blue, orange, and red) with the title 'Sobre o Noosfero'. To the left of this graphic is a sidebar menu with links: 'Entrar', 'Community's profile', 'Blog', 'Agenda', 'Horários', 'Arquivos', 'Galeria de Imagem', 'Features & Issues', 'Resumo de Reuniões', and 'Fórum'. The main text describes the Noosfero platform as a web platform for social and solid economy networks, listing its features like Blog, e-Portfolios, CMS, RSS, etc. It also mentions the 'Noosfero MES-UnB' project, which is a collaborative effort to improve the MES discipline. A 'Traduções' section indicates it is licensed under CC BY. On the right side, there is a 'Sobre o Noosfero' sidebar with links: 'Site Oficial', 'Desenvolvimento', 'Lista de E-mail', 'Easy to Solve', 'Video Tutorials', and 'Convenções'. Below this is a 'MES 01/2014' section with links: 'Dívida Técnica', 'Work Progress', 'Apresentação Final', 'Inception', and 'Informações e Dicas'. At the bottom right, there is a 'Repositórios' section with links: 'Repositório Oficial' and 'Repositório MES'.

Figura 7 – Exemplo do uso do Comunidade.UnB na disciplina de MES.

a comunicação com os seus principais desenvolvedores.

Apesar da Comunidade.UnB estar disponibilizada como um ambiente de testes e não possuir uma vasta divulgação pela Universidade de Brasília, no seu primeiro ano de criação contava com 153 usuários e 14 comunidades. Ao final de maio de 2015 contabilizava 376 usuários e 30 comunidades demonstrando que houve crescimento.

Alguns professores adotaram a Comunidade.UnB como um ambiente de apoio ao Moodle, AVA oficial adotado pela UnB. Na Figura 7 é apresentado um exemplo de seu uso na disciplina de Manutenção e Evolução de Software (MES) ministrada pelo professor Paulo Roberto Miranda Meirelles.

Esse é um exemplo de comunidade criada dentro do Comunidade.UnB, que possui características que se assemelham ao ambiente Moodle, como evidenciado na Seção 5.1, carece de alguns funcionalidades importantes mas com a vantagem da possibilidade de acesso do público ao conteúdo, e a continuidade do conteúdo desenvolvido por outras pessoas que eventualmente se juntem ao longo do tempo. Vale lembrar que em uma publicação de conteúdo os níveis de privacidade podem ser alterados basicamente entre públicos e privados.

As limitações da plataforma e a proposta de uso da Comunidade.UnB por professores para a criação de disciplinas, estimulam o desenvolvimento de funcionalidades que levem à plataforma Noosfero a assemelhar-se a um ambiente virtual de aprendizagem.

Tendo como base todo o processo de desenvolvimento do Noosfero e as principais

vantagens ao se utilizar o Comunidade.UnB, é o que nos motiva realizar a evolução da plataforma. Assim sendo é indispensável direcionar os esforços para a melhor satisfação dos usuários, o que nos leva a comparar no Capítulo 5 os recursos utilizados nos AVA com o Noosfero e propor histórias a serem desenvolvidas.

5 Estudo preliminar

Nesse capítulo é apresentada uma comparação entre o Noosfero e os AVA, assim como a proposta de estudo desse trabalho para a minimização das diferenças entre o Noosfero e os ambientes virtuais de aprendizagem. É evidenciado a pesquisa que forneceu embasamento para direcionar o processo de desenvolvimento, inclusive a descrição das funcionalidades propostas e seus respectivos cenários de uso.

5.1 Comparação entre Noosfero e ambientes virtuais de aprendizagem

Como o objetivo do trabalho é a evolução de plataforma de redes sociais para um ambiente virtual de aprendizagem, a pesquisa realizada fundamenta-se no levantamento das principais funcionalidades que os AVA atuais utilizam, para verificar quais a plataforma o Noosfero ainda não possui. Desse modo surgiu a necessidade do levantamento das principais ferramentas AVA utilizadas atualmente pelas universidades e instituições de ensino, para proporcionar o mapeamento destas funcionalidades. Os AVA selecionados para comparação são **BlackBoard**, **Sakai**, *Modular Object-Oriented Dynamic Learning Environment* (**Moodle** ¹) e *TelEduc*.

O BlackBoard ², é um dos maiores ambientes virtuais de aprendizagem utilizados atualmente por um, total de 72% das TOP 200 melhores universidades do mundo (BLACKBOARD, 2015). Para utilização desta plataforma é necessário a aquisição de licenças por parte da instituição que pretende implementá-la. A plataforma tem total suporte da empresa que a mantém e não permite a modificação de sua estrutura interna pelos seus utilizadores. Os ajustes e melhorias de funcionalidades são realizados através do mapeamento dos comentários e sugestões dos usuários.

O Moodle, é um software livre criado em 2001 pelo educador cientista computacional Martin Dougiamas. É desenvolvido na linguagem de programação PHP e desta maneira pode ser instalado em qualquer sistema operacional que tenha suporte à linguagem. Para o seu uso podem ser utilizadas várias bases de dados com suporte a ODBC³, tais como MySQL, PostgreSQL e Oracle.

O projeto Sakai⁴ ou a comunidade Sakai desenvolve e distribui o software livre Sakai como ambiente virtual de aprendizagem para colaboração e ensino para educadores,

¹ Disponível em: <<https://moodle.org/>>

² Disponível em: <<http://www.blackboard.com/>>

³ Padrão para acesso a sistemas gerenciadores de bancos de dados

⁴ Disponível em: <https://sakaiproject.org/>

por educadores. A plataforma foi desenvolvida em linguagem Java e pode ser executado em várias plataformas diferentes como Linux, Unix, Windows e MAC. Suporta banco de dados MySQL e Oracle. As instituições que utilizam esta plataforma são abrangentes e estão listadas no site da comunidade ⁵.

O **TelEduc**⁶ é um ambiente virtual de aprendizagem, cujo desenvolvimento iniciou em 1997, a partir de uma dissertação de mestrado de Alessandra de Dutra e Cerceau do Instituto de Computação da Universidade Estadual de Campinas⁷. Foi objeto de pesquisa e desenvolvimento de um projeto coordenado pela Profa Dra Heloísa Vieira da Rocha⁸ até 2012. Desde então, o projeto vem sendo evoluído para implementar ajustes e novas funcionalidades segundo (ROCHA; MORAES, 2002), visando implementar necessidades relatadas por seus Usuários. O projeto é mantido pelo Núcleo de Informática Aplicada à Educação ⁹ da UNICAMP.

O TelEduc é um software livre desenvolvido na linguagem de programação PHP ¹⁰ e JavaScript com banco de dados o MySQL ¹¹ para ambientes Linux. Em seu lançamento tornou-se um dos softwares mais utilizados para apoiar a educação à distância nas mais diversas áreas. Seus principais usuários são as universidades públicas e privadas, que utilizam a ferramenta para atividades educacionais, disponibilizando materiais, dando suporte a comunicação e interação entre os participantes.

Foram criadas tabelas comparativas (Tabelas 1, 2, 3, 4, 5) entre as ferramentas selecionadas e a plataforma Noosfero, desse modo fica mais evidente identificar quais funcionalidades a plataforma Noosfero carece. E vale ressaltar que não é objetivo deste trabalho esclarecer ou afirmar qual é a melhor ferramenta, mas verificar as principais funcionalidades dos AVA que são utilizados e aceitos pelas universidades.

As funcionalidades incluídas nas tabelas foram obtidas da literatura apresentada na seção 3.1, e em uso aos ambientes online de demonstração, contando ainda, com o auxílio de mecanismos de ajuda fornecidos pelas ferramentas. Assim sendo, as funcionalidades selecionadas foram divididas em oito categorias:

- **Conteúdo:** relacionadas à criação e manutenção de conteúdos publicados pelos usuários.
- **Atribuição:** permitem que os usuários carreguem arquivos do seu computador local para a plataforma.

⁵ Disponível em: <https://www.sakaiproject.org/community>

⁶ Disponível em: <http://www.teleduc.org.br/>

⁷ IC/UNICAMP. Disponível em: <http://www.ic.unicamp.br/>

⁸ <http://lattes.cnpq.br/6985892121344767>

⁹ NIED/UNICAMP. Disponível em: <http://www.nied.unicamp.br/>

¹⁰ Disponível em: <http://php.net/>

¹¹ Disponível em: <https://www.mysql.com/>

- **Ferramentas:** mecanismos de apoio ao gerenciamento do conteúdo e ambiente;
- **Teste/Quiz:** permitem a realização de avaliações e acompanhamento de resultados dos alunos;
- **Comunicação:** possibilitam a comunicação síncrona e assíncrona entre os participantes de um curso;
- **Curso:** fundamentais para organização dos cursos;
- **Permissão e papéis:** utilizadas para gerenciar o ambiente, definindo papéis e autorizações de acesso ao ambiente;
- **Página principal:** são as principais funcionalidades evidenciadas na página inicial de cada usuário.

Das categorias listadas na Tabela 1, podemos destacar que o Noosfero carece do SCORM ¹² e IMS-Content-Package ¹³ que são funcionalidades importantes para os AVA, pois permitem a importação de conteúdos de outras ferramentas proporcionando flexibilidade para o usuário.

Verifica-se que o Noosfero possui funcionalidades básicas de comunicação, mas não possibilita a comunicação síncrona (Bate-papo). Possui ainda, fóruns que normalmente são utilizados para a divulgação de notícias e dúvidas, entretanto, não permite a assinatura de tópicos para o recebimento de notificações.

Como o Noosfero não é um ambiente virtual de aprendizagem, é possível verificar na Tabela 4 que ele não possibilita a criação de questões para obter a resposta de usuários. Também não conta com a funcionalidade que permita ao professor atribuir as notas e acompanhar a situação de cada aluno. O Noosfero possui um *plugin* denominado *Work Assignment* que permite aos usuários enviarem arquivos para o servidor mas não possibilita que o moderador estabeleça uma pontuação para os mesmos.

¹² Um modelo de referência seja, conjunto unificado de especificações para a disponibilização de conteúdos e serviços de e-learning (DUTRA; TAROUÇO, 2006).

¹³ Padrão que permite exportar o conteúdo de um sistema de gerenciamento de conteúdo de aprendizagem ou repositório digital.

Funcionalidades\Sistema	BlackBoard 9.1	Moodle	TelEduc	Sakai	Noosfero
Conteúdo					
Diretórios/Pastas	Sim	Sim	Sim	Sim	Sim
Criar/Extrair arquivos de pastas	Sim	Sim	Sim	Sim	Sim
Editor HTML	Sim	Sim	Sim	Sim	Sim
Filtro de vários idiomas	Não	Sim	Não	Não	Sim
Audio	Sim (incorporado no plugin do QuickTime)	Sim	Não	Sim	
Vídeo	Sim	Sim	Não	Sim	Sim
SCORM	Sim	Sim	Não	Sim	Não
IMS-Content-Package	Sim	Sim			Não
Atribuição					
Upload de um arquivo	Sim	Sim	Sim	Sim	Sim
Upload de vários arquivos	Sim	Sim	Não	Sim	Sim
Texto online	Sim(quiz)	Sim	Sim	Sim	Sim
Comunicação					
Enviar e-mail	Sim	Sim	Sim	Sim	Sim
Enviar mensagens	Sim	Sim	Sim	Sim	Sim (mural)
Bate papo	Sim	Sim	Sim	Sim	Não
Fórum de discussão	Sim	Sim	Sim	Sim	Sim
Alterar tipo de fórum	Sim	Sim	Sim	Sim	Sim
Fórum:	Sim	Sim	Sim	Sim	Sim
- Postagem anônima	Sim	Não	Não	Não	Não
- anexar um arquivo	Sim	Sim	Não	Sim	Não
- os participantes podem criar tópicos	Sim	Sim	Não	Sim	Sim
- inscrever-se em um fórum	Sim	Sim	Não	Sim	Não
- assinar tópicos	Sim	Não	Não	Sim	Não
- moderar fórum	Sim	Não	Sim	Sim	Não
Pesquisar fóruns	Sim	Sim	Sim	Sim	Sim (busca geral)

Tabela 1 – Tabela de comparação categorias: Conteúdo e Atribuição

Funcionalidades\Sistema	BlackBoard 9.1	Moodle	TelEduc	Sakai	Noosfero
Curso					
Criar cursos	Sim	Sim	Sim	Sim	Sim
Criar novos papéis	Sim	Sim	Não	Não	Sim
Ferramentas de grupos	Sim	Sim	Sim	Sim	Sim
Relatórios do curso	Sim	Sim	Sim	Sim	Não
Sistema de alerta	Sim	Não	Não	Não	Não
Personalização ou configuração (nome, duração, inscrição, idioma)	Sim	Sim	Sim	Sim	Sim
Formato de curso(fórum, formato dos tópicos, semanal)	Não	Sim	Não		Não
Gerenciar campo menu	Sim	Sim O menu é personalizado de acordo com conteúdo o curso	Não	Sim	Sim
Customizar estilo de curso	Sim (menu, tema)	Sim (tema)	Não		Sim
Estrutura de pastas	Sim	Sim	Não	Sim	Sim
Gerenciar ferramentas	Sim	Sim	Sim	Sim	Sim
Backup do curso / exportação	Sim	Sim	Não	Sim	Sim
Importar curso	Sim	Sim	Não	Não	Não
Restaurar curso	Sim	Sim	Não	Não	Não
Importar conteúdo/características de outros cursos	Sim	Sim	Sim	Sim	Sim
Gerenciador de arquivos	Sim	Sim	Não	Sim	Sim

Tabela 2 – Tabela de comparação categoria Curso

Funcionalidades\Sistema	BlackBoard 9.1	Moodle	TelEduc	Sakai	Noosfero
Ferramentas					
Glossário	Sim (Editável pelo instrutor)	Sim (Os participantes também podem adicionar entradas) + bloco para glossário	Não	Sim	Não
RSS Feeds	Não	Sim (Banco de dados, Fórum, Glossário)	Não	Sim	Não
Blogs	Sim	Sim	Sim	Sim	Sim
Lista de contatos	Sim	Sim	Sim	Não	Sim (lista de participantes)
Calendário (Curso)	Sim	Sim	Sim	Sim	Apenas de eventos
Permissões de acesso	Sim (programada, dependendo de grau, dependendo do nível total do curso, depende do acesso a conteúdo definido, definir suas próprias regras)	Sim (programada, dependendo de grau, dependendo do nível total do curso, depende do acesso a conteúdo definidos / atividades)	Sim	Sim	Sim
Verificação de plágio	Sim (integrado)	Não (Não integrado, mas existem APIs)	Não	Não	Não
Busca de arquivos (curso)	Sim (adicionando um bloco correspondente)	Não	Sim	Sim	Sim
Jornal	Sim	Não	Não		Não
Atividade concluída	Sim	Sim	Sim	Sim	Sim (Não totalmente implementado)
Ferramentas de voz	Não (é necessário a ferramenta Wimba instalada)	Não	Não	Sim	Não
Permite o uso de módulos e plugins	Sim	Sim	Não	Sim (permite a seleção de ferramentas)	Sim

Tabela 3 – Tabela de comparação categoria Ferramentas

Funcionalidades\Sistema	BlackBoard 9.1	Moodle	TelEduc	Sakai	Noosfero
Teste/Quiz					
Banco de questões	Sim	Sim	Sim	Sim	Não
Exportar resultados	Sim	Sim	Sim	Não	Não
Exportar respostas	Sim	Não	Não	Sim	Não
<i>Tipo de questões</i>					
resposta com arquivo	Sim	Sim	Não	Sim	Sim
resposta por escala	Sim	Sim (múltipla escolha)	Não	Não	Não
preencher espaço vazio	Sim	Sim (difícil de lidar)	Não	Sim	Não
múltipla escolha	Sim	Sim	Sim	Sim	Não
resposta múltipla	Sim	Sim	Sim	Sim	Sim (permite o envio de várias versões de um arquivo)
Verdadeiro/Falso	Sim	Sim	Sim	Sim	Não
Calculos	Sim	Sim	Não	Sim	Não
Resposta dissertativa	Sim	Sim	Sim	Sim	Sim (com a criação de um artigo)
Calculo numérico	Sim	Sim	Não	Sim	Não

Tabela 4 – Tabela de comparação categoria Teste/Quiz

Funcionalidades\Sistema	BlackBoard 9.1	Moodle	TelEduc	Sakai	Noosfero
Permissões e papéis					
Papéis pré-definidos	Sim (instrutor, assistente de ensino, construtor curso, nivelador, estudante, convidado;)	Sim (Administrador do site, gerente, criador do curso, professor editor, professor, estudante, convidado, usuário;)	Sim (coordenadores, formadores, alunos, visitantes, colaboradores;)	Sim (instrutores, assistente de professor, estudante;)	Sim (Administrador do sistema, moderador, membro, administrador de perfil;)
Editar papéis existentes	Sim	Sim	Não	Não	Sim (apenas com permissão)
Criar novos papéis	Sim	Sim	Não	Não	Sim
Página principal					
RSS Feeds	Não	Sim	Não	Sim	Não
Avisos/ Novos fóruns	Sim	Sim	Sim	Sim	Não
Lista de afazeres/ Próximos eventos	Sim	Sim	Sim (Agenda)	Sim	Sim (Calendário)
Mensagens	Não	Sim	Sim	Sim	Sim
Boletim/ Resultados de questionários	Sim	Sim	Sim	Sim	Não
Usuários online	Sim	Sim	Sim	Sim (chat)	Não
Meu calendário	Sim	Sim	Sim	Sim	Sim
Baixar o conteúdo completo do curso	Não	Não	Sim (todos os arquivos)	Sim	Não
Informação pessoal/ Perfil de usuários	Sim	Sim	Sim	Sim	Sim
Cancelar a inscrição de um curso	Sim	Sim	Não	Sim	Sim
Software livre - GPL license	Não	Sim	Sim	Sim	Sim
Extensão a plataformas móveis	Sim	Não	Não	Não	Não

Tabela 5 – Tabela de comparação categorias: Permissões e papéis, página principal

Os AVA mencionados possuem papéis de usuários pré-estabelecidos (Tabela 5), apesar do Noosfero possuir estes papéis preestabelecidos possui a vantagem de criação de papéis customizados, onde o administrador define quais as permissões para cada um deles.

O Noosfero carece de uma melhor estruturação dos conteúdos para o gerenciamento de um curso. Conforme evidenciado na Tabela 2, a plataforma não permite tratar os conteúdos na forma de cursos e tampouco a personalização de seus formatos (formato dos tópicos, duração). Além disso a ferramenta de calendário disponível mostra apenas as datas dos eventos criados, não disponibilizando aos alunos todas as atividades e duração do curso de forma limpa e organizada.

Quando analisados os itens da categoria de página principal (Tabela 5) é verificado que o Noosfero necessita de um template próprio para cursos que reúne as informações principais dos alunos. Algumas dessas informações necessitam apenas de uma melhor disposição pela interface, como calendário e mensagens. Já outras exigem implementação como um sistema de avisos, usuários online e boletim.

Analisando todas as Tabelas elaboradas fica evidente que é essencial ao Noosfero um mecanismo para atribuição de notas e melhor controle sob o envio de tarefas. Esses são os pontos principais que regem um ambiente virtual de aprendizagem além das funcionalidades relatadas.

5.2 Pesquisa com alunos

Para melhor prosseguirmos com este trabalho, realizamos uma pesquisa junto aos alunos da FGA sobre o uso de redes sociais em seu dia a dia e sobre o seu uso para apoio as disciplinas cursadas na universidade. Para isso, foi criado um questionário com quatro questões de múltipla escolha, e para os itens associou-se a uma escala tipo *Likert* de cinco pontos: Nunca; Raramente; Às vezes; Frequentemente; Sempre (Instruções de preenchimento no Apêndice A). Com este tipo de escala, pode-se mapear todas as respostas seguindo a lógica em que há duas alternativas negativas, duas alternativas positivas e uma intermediária.

Nesta seção, apresentamos as questões elaboradas (Apêndice B) e as respostas coletadas através da ferramenta *Google Forms*. O questionário foi aplicado na rede social *Facebook*, em grupos específicos da FGA. Foram coletadas, ao todo noventa e quatro respostas voluntárias ao questionário. Abaixo segue as questões com as respectivas análises de seus resultados, que estão representados graficamente no Apêndice ??.

Com que frequência você utiliza redes sociais?

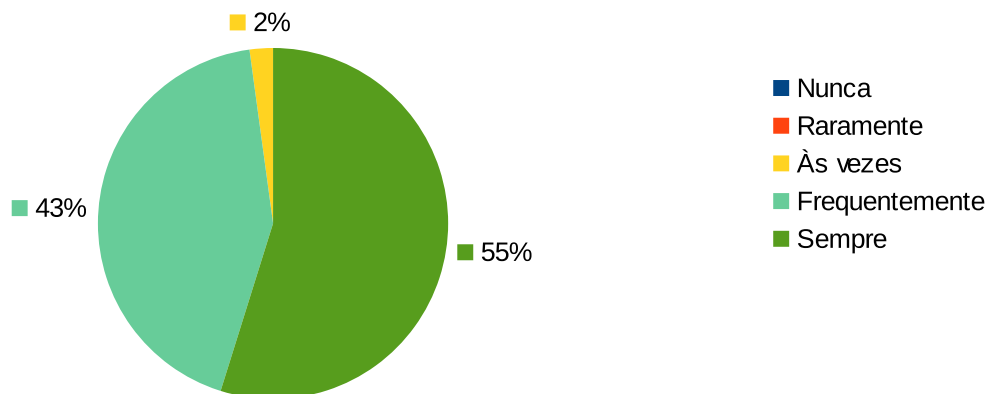


Figura 8 – Resultados do questionário para a pergunta 1

Com que frequência você utiliza redes sociais?

A primeira pergunta tinha como objetivo levantar com que frequência os alunos utilizam as redes sociais. Na Figura 8, nota-se que 98% dos alunos que responderam ao questionário utilizam as redes sociais sempre ou frequentemente, o que evidencia o fato que atualmente os estudantes dessa geração estão inseridos neste contexto.

Você utiliza alguma rede social como ferramenta de apoio as disciplinas?

Você utiliza alguma rede social como ferramenta de apoio as disciplinas?

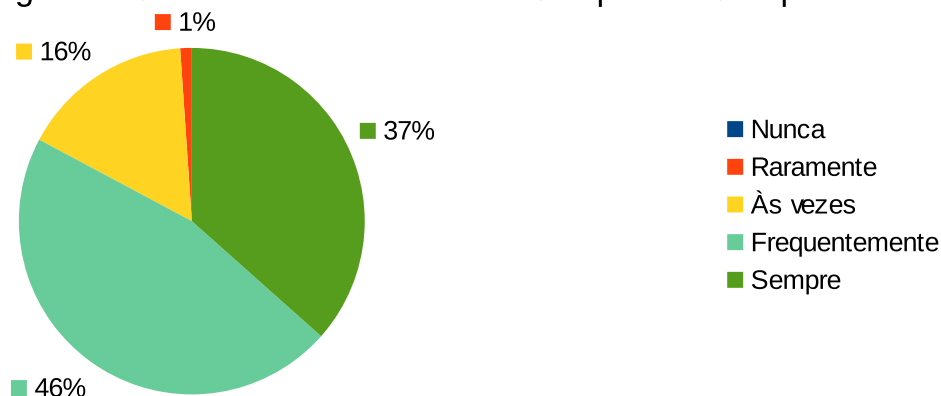


Figura 9 – Resultados do questionário para a pergunta 2

O intuito da segunda pergunta foi verificar se os alunos utilizam alguma ferramenta de redes social como ferramenta de apoio as disciplinas. Na Figura 9, verifica-se que apenas 16% dos alunos às vezes utilizam e de outro ponto de vista 83% dos alunos fazem uso da rede para tal. Demonstrando que mesmo a universidade utilizando AVA os alunos se

apoiam em redes sociais para discutir e compartilhar conteúdos referentes as disciplinas cursadas.

Os professores incentivam o uso de redes sociais para suas disciplinas?

Os professores incentivam o uso de redes sociais para suas disciplinas?

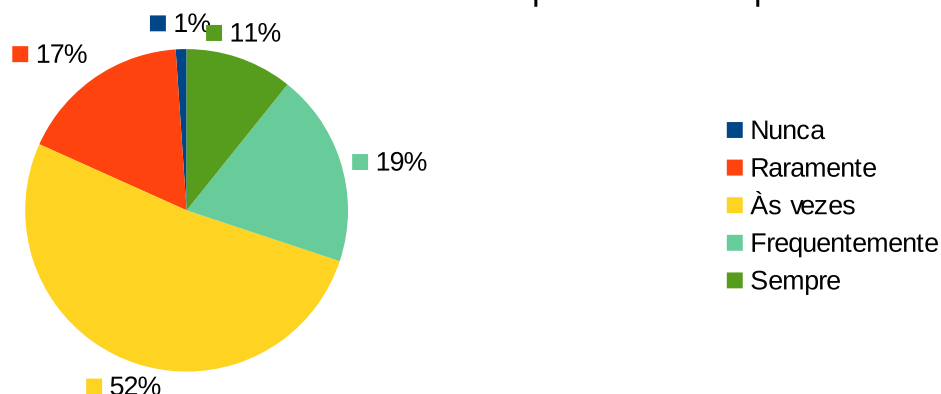


Figura 10 – Resultados do questionário para a pergunta 3

O objetivo desta pergunta foi verificar se mesmo que a universidade indique o uso de AVA se os professores incentivam o uso das redes sociais em suas disciplinas. Na Figura 10, percebe-se que pouco mais da metade dos alunos (52 %) tem suas respostas em um ponto intermediário do questionário, mostrando que os professores são bem imparciais quanto a isso. Apesar que do restante, tem-se mais respostas favoráveis do que contra.

Mesmo que o professor não recomende o uso de redes sociais para discussão de conteúdos de suas disciplinas, você as utiliza?

Mesmo que o professor não recomende o uso de redes sociais para discussão de conteúdos de suas disciplinas, você as utiliza?

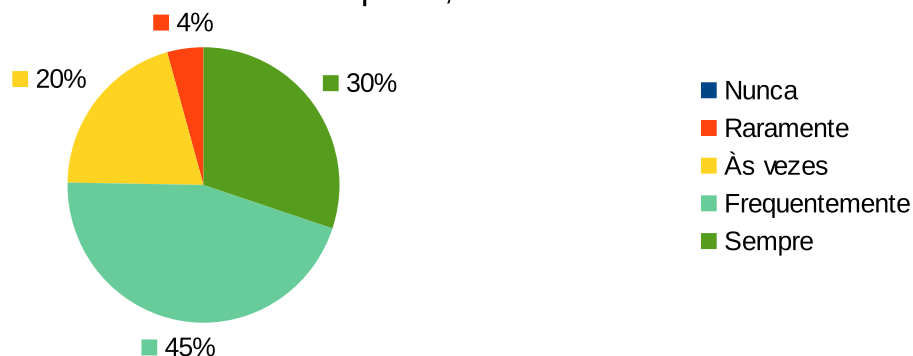


Figura 11 – Resultados do questionário para a pergunta 4

Na quarta e última pergunta buscou-se investigar se os alunos utilizam as redes sociais como uma ferramenta de apoio as disciplinas mesmo que os professores não recomende seu uso para tal. Dos que responderam 75% não levam em consideração tais recomendações e fazem o uso, demonstrando que os alunos preferem estar nas redes sociais onde expõem suas opiniões a fim de promover o compartilhamento de conteúdo (Resultados na Figura 11).

Esta pesquisa exploratória mostra que mesmo com a existência de AVA os alunos utilizam as redes sociais, que não foram criadas com esse objetivo, para a discussão de assuntos relacionados ao ambiente acadêmico. Fundamentado nesses resultados e na comparação desses ambientes com o Noosfero, foram elaboradas funcionalidades para minimizar a diferença encontrada.

5.3 Histórias de usuário

Nesta seção apresentaremos as funcionalidades desenvolvidas para contribuir para a adequação do Noosfero a um ambiente virtual de aprendizagem, para exercer a função de apoio aos AVA adotados pela Universidade. Serão apresentados os requisitos funcionais levantados para que esta possa suprir as necessidades levantadas.

A principais funcionalidades estão relacionadas ao *plugin WorkAssignmentnet*, além disso, julgou-se necessária a evolução do *plugin Comunidade.UnB*. Este foi iniciado por Bucher (2013), com objetivo de integrar o Noosfero com o serviço de Lightweight Directory Access Protocol (LDAP). Uma base LDAP é utilizada pela UnB para manter os dados de todos os seus alunos e funcionários.

As funcionalidades são apresentadas utilizando o utilizando o formato de histórias de usuários¹⁴, de acordo com as práticas ágeis. Além disso, os critérios de aceitação estão expostos no formato de cenários de uso, utilizando BDD.

Plugin WorkAssignment

Nesta subseção serão apresentadas um primeiro levantamento das histórias correspondentes a evolução do plugin Work Assignment. Até então, o plugin possui apenas a funcionalidade de permitir o envio de arquivos para o servidor em um determinado período de tempo.

A proposta de evolução é a criação de um sistema de notas que permita ao professor atribuir notas as atividades enviadas e dessa maneira acompanhar a situação de cada aluno. Do ponto de vista do aluno, o mesmo poderá visualizar todas as notas das atividades em cada disciplina, avaliando se o seu desempenho está satisfatório.

¹⁴ Em inglês *User Stories*(US)

Como avanço evoluir o *plugin* de tal forma que seja oferecido ao professor definir o tempo em que a atividade estará disponível para o envio, uma vez que atualmente é realizada de maneira manual, não possibilitando definir intervalos de tempo. A seguir estão descritas as histórias correspondentes a essa evolução.

US01 Definir tempo restante

Como um professor

Gostaria de definir o tempo restante para cada atividade no Work Assignment

Para gerenciar o envio de atividades pelos alunos em um determinado período.

Cenários de uso:

1. Definir tempo

[**Dado**] que esteja logado como professor

[**E**] selecione a opção “Gerenciar conteúdo”

[**Quando**] eu clicar em trabalho a ser entregue

[**Então**] devo visualizar a opção “Ativar tempo de entrega”

[**E**] informar a data e hora máxima para envio da atividade.

2. Modificar tempo

[**Dado**] que esteja logado como professor

[**E**] eu tenha atividades a ser entregue em aberto

[**E**] eu esteja visualizando as informações da atividade

[**Quando**] eu selecionar opção “Editar”

[**E**] visualizar a opção “Tempo de entrega”

[**Então**] deve ser permitido que eu altere o tempo definido

[**E**] visualize o resultado da alteração após a confirmação.

3. Permitir entrega de atividades após período

[**Dado**] que esteja logado como professor

[**E**] selecione a opção “Gerenciar conteúdo”

[**E**] navegar até a página trabalho a ser entregue

[**Quando**] selecionar a opção “Ativar tempo de entrega”

[**Então**] devo visualizar a opção “Permitir entrega após o período”

[**Para**] que eu possa selecioná-la.

US02 Visualizar tempo restante de atividades

Como um aluno

Gostaria de visualizar o tempo restante da atividade

Para enviar um arquivo na data correta.

Cenários de uso:

1. Visualizar tempo

[**Dado**] que esteja logado como Aluno

[**E**] eu esteja inscrito em um Curso

[**E possui atividade em aberto**]

[**Quando**] eu selecionar a opção de visualizar a atividade

[**Então**] eu devo visualizar se a atividade está em aberto

[**E**] qual o tempo restante para o envio da atividade.

US03 Atribuir notas aos alunos

Como um professor

Gostaria de atribuir notas as atividades enviadas pelos alunos

Para avaliar o rendimento de cada um deles.

Cenários de uso:

1. Atribuir notas

[**Dado**] que esteja logado como professor

[**E**] que a funcionalidade de notas esteja habilitada no plugin

[**Quando**] selecionar uma atividade na lista de atividades

[**Então**] devo visualizar a opção atribuir nota

[**E**] devo visualizar o campo “nota” para preenche-lo.

2. Alterar notas

[**Dado**] que esteja logado como professor

[**E**] que a funcionalidade de notas esteja habilitada no plugin

[**Quando**] selecionar uma atividade na lista de atividades

[**E**] atividade já possua notas

[**Então**] devo visualizar a opção alterar nota

[**E**] devo visualizar o campo “nota” para preenche-lo.

3. Definir critério da nota final

[**Dado**] que esteja logado como professor

[**E**] que a funcionalidade de notas esteja habilitada no plugin

- [Quando] criar um trabalho a ser enviado
- [Então] devo visualizar a opção de qual o critério para nota final
- [E] devo selecionar qual a opção desejada.

US04 Publicar notas aos alunos

Como um professor

Gostaria de publicar as notas atribuídas as atividades

Para que os alunos possam visualizar suas respectivas avaliações

Cenários de uso:

1. Disponibilizar notas de uma determinada atividade
 - [Dado] que esteja logado como professor
 - [E] que a funcionalidade de notas esteja habilitada no plugin
 - [Quando] selecionar uma atividade na lista de atividades
 - [Então] devo visualizar a opção “permitir visualização”
 - [E] devo ter a opção de ativá-lo.
2. Omitir notas de uma determinada atividade
 - [Dado] que esteja logado como professor
 - [E] que a funcionalidade de notas esteja habilitada no plugin
 - [Quando] selecionar uma atividade na lista de atividades
 - [Então] devo visualizar a opção “permitir visualização”
 - [E] devo ter a opção de desativá-lo.

US05 Visualização das notas

Como um aluno

Gostaria de visualizar minhas notas

Para inteira-me sobre minha pontuação nas atividades.

Cenários de uso:

1. Visualizar cursos com notas disponíveis
 - [Dado] que estou logado como “Hebert”
 - [E] o plugin Work Assignment esteja ativo
 - [E] participo de alguma comunidade que utilize o plugin de notas
 - [E] existem notas de atividades disponíveis
 - [E] o professor tenha permitido sua visualização

[Quando] eu navegar até a página “Minhas notas”

[Então] tenho que ver todos as comunidades (cursos) com notas disponíveis.

2. Detalhar notas de cada curso

[Dado] que estou logado como “Hebert”

[E] o plugin Work Assignment esteja ativo

[E] participo de alguma comunidade que utilize o plugin de notas

[E] existem notas de atividades disponíveis

[E] o professor tenha permitido sua visualização

[E] navegar até a página “Minhas notas”

[E] visualizar todos as comunidades (cursos) com notas disponíveis

[Quando] eu selecionar a “Visualizar detalhes” de alguma comunidade

[Então] tenho que ver todos os grupos de atividades

[E] as notas disponibilizadas pelo professor.

US06 Professor gerencia notas

Como um professor

Gostaria de gerenciar as notas dos integrantes da comunidade

Para manter o controle sobre a pontuação de todos os alunos.

Cenários de uso:

1. Definir grupo de atividades

[Dado] que esteja logado como professor

[E] que a funcionalidade de notas esteja habilitada no plugin

[E] selecionar a opção “Gerenciar notas”

[E] e eu selecionar o curso desejado

[Quando] eu clicar em “Cadastrar grupo de atividades”

[E] eu preencho os campos

“Nome do grupo”,

e “Lista de Atividades”

[E] eu clico em “Salvar”

[Então] recebo uma mensagem de confirmação

[E] visualizo todas os grupos de atividades criados.

2. Visualizar notas de todos os alunos de uma determinada atividade

[Dado] que esteja logado como professor

[E] que a funcionalidade de notas esteja habilitada no plugin

- [E] selecionar a opção “Gerenciar notas”
 - [E] e eu selecionar o curso desejado
 - [Quando] eu selecionar atividade
 - [E] algum aluno tenha enviado a atividade
 - [Então] devo visualizar todas as atividades enviadas e suas respectivas notas.
3. Visualizar notas de todos os alunos de um grupo de atividades
- [Dado] que esteja logado como professor
 - [E] que a funcionalidade de notas esteja habilitada no plugin
 - [E] selecionar a opção “Gerenciar notas”
 - [E] e eu selecionar o curso desejado
 - [Quando] eu clicar em “Visualizar notas por grupo de atividades”
 - [E] algum aluno tenha enviado a atividade
 - [Então] devo visualizar todas as atividades daquele grupo e suas respectivas notas.

US07 Bloco de notas recentes

Como um aluno

Gostaria de visualizar minhas cinco últimas notas

Para facilitar a visualização do meu desempenho no curso.

Cenários de uso:

1. Adicionar bloco
 - [Dado] que esteja logado como aluno
 - [Quando] eu navegar até meu painel de controle
 - [E] eu selecionar a opção “Editar Blocos Laterais”
 - [E] eu clicar na opção “Adicionar bloco”
 - [Então] devo visualizar a opção “Notas Recentes”
 - [E] devo ter a opção de adicioná-lo.
2. Visualizar cinco notas recentes
 - [Dado] que esteja logado como aluno
 - [E] esteja com o bloco de notas recentes adicionado
 - [E] o professor tenha publicado alguma nota
 - [Então] devo visualizar o bloco com minhas cinco últimas notas
 - [E] o nome de cada atividade.

US08 Bloco para visualização dos módulos

Como um professor

Gostaria de visualizar um bloco com todos os módulos disponíveis no meu curso

Para facilitar a visualização da organização do curso criado.

Cenários de uso:

1. Adicionar bloco

[**Dado**] que esteja logado como professor

[**E**] e seja administrador da comunidade

[**Quando**] eu navegar até painel de controle da comunidade

[**E**] eu selecionar a opção “Editar Blocos Laterais”

[**E**] eu clicar na opção “Adicionar bloco”

[**Então**] devo visualizar a opção “Lista de Grupos de trabalhos a serem enviados”

[**E**] devo ter a opção de adicioná-lo.

2. Editar numero de atividades listadas no bloco

[**Dado**] que esteja logado como professor

[**E**] e seja administrador da comunidade

[**Quando**] eu navegar até painel de controle da comunidade

[**E**] eu selecionar a opção “Editar Blocos Laterais”

[**E**] eu clicar na opção “Editar bloco”

[**Então**] devo visualizar a opção “Limite de trabalhos enviados”

[**E**] permitir escolher o número de itens a serem visualizados.

3. Visualizar módulos

[**Dado**] que esteja logado como aluno

[**E**] seja membro da comunidade

[**E**] a comunidade possua grupos de atividades a serem enviadas

[**E**] possua trabalhos a serem enviados

[**Então**] devo visualizar o bloco todos os módulos disponíveis

[**E**] as atividades relacionadas a cada um deles.

Plugin Comunidade.UnB

O objetivo do *plugin* do Comunidade.Unb é que o usuário tenha acesso através dos mesmos dados utilizados para acessar outros serviços, como, por exemplo, o serviço de matrícula ¹⁵ para alunos ou o serviço de lançamento de notas para professores.

A versão anterior do *plugin* permitia que apenas os novos usuários realizassem o cadastro de suas matrículas, dessa maneira aqueles que já estavam cadastrados não tinham acesso ao portal. Devido às limitações de acesso definidas pelo *plugin*, onde apenas usuários com matrícula cadastrada tem acesso ao sistema. A seguir estão descritas as histórias correspondentes.

US09 Autenticação via LDAP

Como um aluno da Universidade Brasília

Gostaria de me autenticar na rede através da minha matrícula e senha

Para utilizar os mesmos dados de cadastro da UnB.

Cenários de uso:

1. Acesso sem matrícula cadastrada

[**Dado**] que sou aluno da UnB

[**E**] possuo cadastro ativo na base de dados da UnB

[**E**] possuo cadastro ativo na base de dados do Comunidade.UnB

[**Quando**] eu acessar o portal

Como um aluno da Universidade Brasília

[**Então**] devo ser direcionado para uma página com o título “Cadastrar Matrícula”

[**E**] devo ver os campos “Matrícula”, “Senha” e “confirmação de senha” em branco.

2. Registro de matrícula de usuários existentes

[**Dado**] que sou aluno da UnB

[**E**] me encontro na página de Cadastrar matrícula do Comunidade.UnB

[**Quando**] eu preencher os campos

“matrícula” com “100103979”,

e “senha” com a senha da base dados da UnB,

e “confirmação de senha”

[**E**] clicar no botão “Registrar”

[**Então**] eu devo ser direcionado para meu perfil

¹⁵ Disponível em: <<https://matriculaweb.unb.br>>

Neste capítulo foi apresentado as funcionalidades de um AVA em comparação com o Noosfero, bem como uma pesquisa realizada com os alunos para direcionar o desenvolvimento do trabalho. Este estudo é importante, uma vez que considerando esta análise, foi apresentado Seção 5.3 as funcionalidades que foram implementadas no desenvolvimento prático deste trabalho.

6 Desenvolvimento de uma Rede Social de Colaborações e Aprendizagem

Neste capítulo será apresentado o processo para a implementação das funcionalidades apresentadas na Seção 5.3, mostrando os passos de desenvolvimento do *plugin WorkAssignment* e Comunidade.UnB para o Noosfero, bem como os resultados obtidos.

6.1 O processo de desenvolvimento

O desenvolvimento foi realizado no Laboratório de Produção, Pesquisa e Inovação em Software (LAPPIS) da FGA que é um ambiente para alunos e professores trabalharem de forma colaborativa na produção de software. No laboratório são adotadas práticas ágeis onde vários de seus integrantes colaboram com o Noosfero.

A partir dos requisitos especificados e validados e levando em consideração as práticas ágeis utilizadas no LAPPIS foram realizadas *sprints* com duração de quinze dias. Em cada *sprint* foram desenvolvidas histórias de acordo com a pontuação e demanda de atividades na equipe do Portal, visto que o desenvolvimento foi realizado de maneira colaborativa.

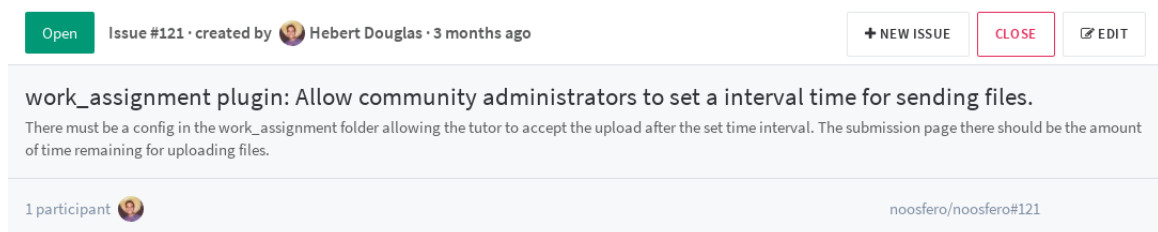


Figura 12 – Issue 121: Gerenciamento de tempo para work_assignment

A primeira *sprint* de desenvolvimento das histórias de usuário foi iniciada em julho com a equipe do Portal FGA. Seguindo as práticas de desenvolvimento da comunidade Noosfero, criou-se uma *Issue* para evidenciar aos desenvolvedores o início da implementação de uma nova funcionalidade. A *Issue* criada foi a de número 121¹ que é evidenciada na Figura 12.

Nessa *sprint* foi desenvolvida a história de usuário (US01) “Definir tempo restante”, que permite ao administrador definir e modificar um tempo limite para envio dos trabalhos no *plugin work assignment*.

¹ Disponível em: <<https://gitlab.com/noosfero/noosfero/issues/121>>

Na segunda sprint, foi desenvolvido o último cenário de uso da **US01**, “Permitir entrega de atividades após período”. Este cenário permite ao professor autorizar os alunos a enviarem suas atividades após o prazo definido. Além de implementar a história “Visualizar tempo restante de atividades” (**US02**), que trata o modo de visualização do tempo restante para envio das atividades.

É importante ressaltar que na implementação das histórias **US01** e **US02** tem-se um exemplo prático de utilização dos *hotspots* da arquitetura do Noosfero. No código 6.1, faz-se uso do *hotspot* denominado como *content_remove_upload*, utilizado para habilitar ou desabilitar a opção de upload de arquivos. Este código está contido no plugin e através dos métodos *dispatch* é invocado pela classe responsável por gerenciar os *plugins*.

Listing 6.1 – Código de implementação do *hotspot*

```
def content_remove_upload(content)
  if content.kind_of?(WorkAssignmentPlugin::WorkAssignment)
    !content.profile.members.include?(context.send(:user))
    || (content.expired? && !content.ignore_time)
  end
end
```

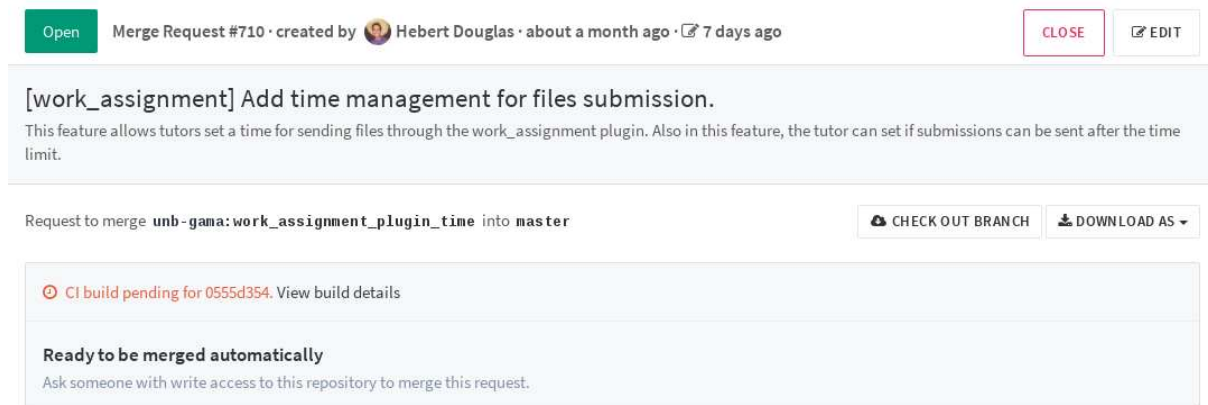


Figura 13 – Merge Request: Gerenciamento de tempo para *work_assignment*

A implementação foi revisada por um dos integrantes do LAPPIS que sugeriu melhorias pontuais, resolvidas durante a revisão. Posteriormente foi realizado o *Merge Request* para a integração do código à *branch master* do Noosfero. Criou-se o Merge Request de número 710² (Figura 13) que ficou em aberto para a revisão dos colaboradores oficiais da Comunidade. Após alguns dias o mesmo foi revisado pelos desenvolvedores do core do Noosfero, que aprovou a funcionalidade e indicou que será integrada ao Noosfero na versão 14.

² Disponível em: <https://gitlab.com/noosfero/noosfero/merge_requests/710>

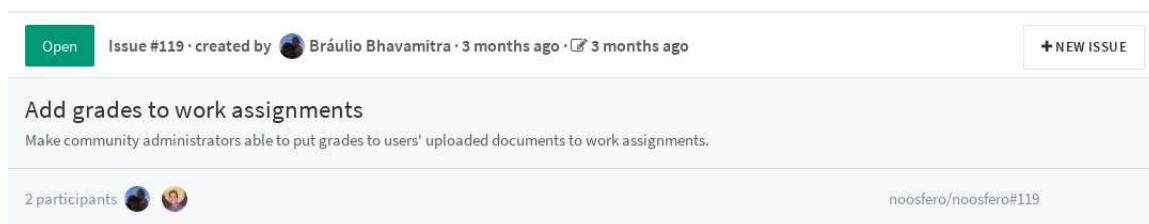


Figura 14 – Issue 119: Adicionar notas ao *plugin Work Assignment*

A terceira *sprint*, que engloba a história (US03) “Atribuir notas aos alunos”. No início da implementação verificou-se que esta funcionalidade já havia sido mapeada por integrante do Noosfero na *Issue* 119 ³ (Figura 14), o que favoreceu o desenvolvimento, e evidenciou que o desenvolvimento na comunidade é feito de maneira colaborativa.

Também foram discutidas questões técnicas e arquiteturais relativas à implementação. As principais discussões giraram em torno das formas de realização da pontuação (por versões ou por pasta) e de atribuição da nota final do aluno. Ficou definido que todas as versões de arquivos enviados poderiam ser pontuados pelos administradores da comunidade e que em uma próxima *sprint* seriam estabelecidos os critérios para definir a nota final do aluno.

Na quarta *Sprint*, prosseguiu-se com o desenvolvimento da US03, momento em que foi implementado o cenário de uso que proporciona ao administrador definir o critério para a nota final de cada atividade. Além disso, foi implementada a história (US04) “Publicar notas aos alunos”, que permite ao administrador publicar ou omitir notas aos alunos.

As funcionalidades desenvolvidas até então englobavam requisitos importantes de um AVA, conforme a Seção 3.1, mas ainda não possibilitam visualização das notas pelo aluno.

Sendo assim na quinta *sprint* foi implementada a história (US05) “Visualização das notas”, restrita à apresentação da tela de visualização das tarefas enviadas. Para que os membros da equipe realizassem uma avaliação relativas ao *design*.

Foi implementada a história (US06) “Professor gerencia notas”, que permite ao professor definir módulos que agrupem as atividades a serem enviadas. O objetivo principal dessa história é permitir ao administrador modularizar as atividades. O cenário “Visualizar notas de todos os alunos de um grupo de atividades” não foi implementado, em vista do elevado número de requisições ao banco de dados e pequeno retorno ao usuário.

Como dívida técnica da quinta, na sexta *sprint* foi implementado o cenário de uso “Visualizar notas de todos os alunos de uma determinada atividade” da US06. A ideia principal é disponibilizar ao usuário, de maneira centralizada, a visualização de suas notas

³ Disponível em: <<https://gitlab.com/noosfero/noosfero/issues/119>>

relativas a todas as atividades e módulos das comunidades nas quais está vinculado.

Na sétima *sprint* foi implementada a história (US07) “Blocos de notas recentes”, que visa a criação de um bloco e exibe ao usuário as cinco notas recentes. Devido as sugestões de membros da comunidade houve uma evolução e padronização de telas do sistema.

Para oitava *sprint* foi sugerida a criação de um bloco para visualização de todos os grupos e respectivas atividades. Esta funcionalidade está listada na US08 (“Bloco para visualização dos módulos”) e foi implementada em paralelo com os testes da história anterior.

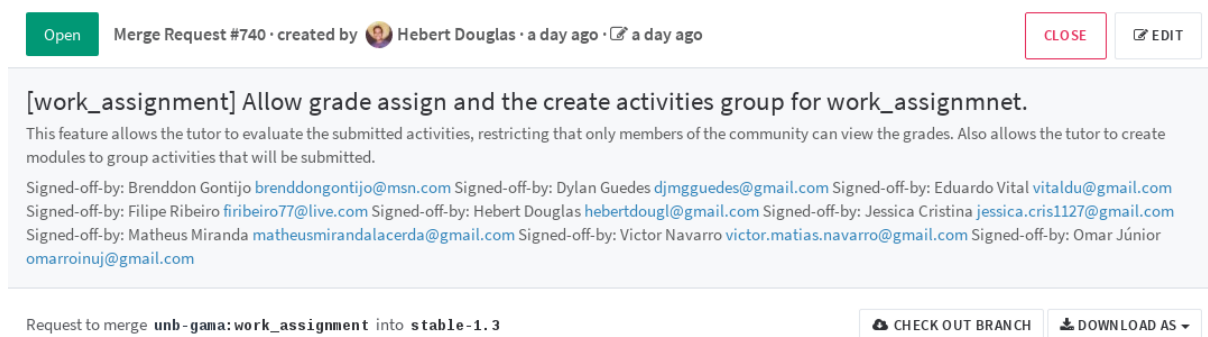


Figura 15 – Merge Request: Sistema de notas e módulos para work_assignment

Finalizada a execução dessas *sprints*, foi realizado o *Merge Request* para a integração do código à *branch master* do Noosfero. Criou-se o Merge Request de número 740⁴ (Figura 15) que ficou em aberto para a revisão dos *commiters* oficiais da Comunidade.

Na nona e última *sprint* foi criado um ambiente de homologação para o Comunidade.UnB, com vistas a propiciar o teste das novas funcionalidades e a autenticação com o LDAP, relatados na US09. Para validar o novo mecanismo de autenticação o CPD da FGA forneceu acesso a base de dados do seu servidor LDAP, apenas uma cópia de uma versão antiga a utilizada para o controle de acesso a rede *wireless* do campus, e durante os testes no servidor de homologação a funcionalidade teve o comportamento esperado.

Nesse processo de desenvolvimento fica evidente a aplicação das práticas adotadas pela comunidade de software livre Noosfero. Foi executado o processo desde a criação de uma *Issue*, passando por todo o processo de desenvolvimento adotando práticas ágeis, até o encerramento do ciclo com a criação do *Merge request*. Vale lembrar que mesmo após a revisão e integração ao *core* o código ainda se encontra em constante evolução.

⁴ Disponível em: <https://gitlab.com/noosfero/noosfero/merge_requests/740>

6.2 A evolução do *plugin Work Assignment*

Nesta seção serão apresentados os resultados correspondentes à evolução do *plugin Work Assignment*, que até então possuía apenas a funcionalidade para permitir o envio de arquivos para o servidor em um determinado período de tempo.

O professor deve atribuir notas às atividades enviadas e acompanhar a situação de cada aluno. O aluno poderá visualizar todas as notas das atividades em cada disciplina, avaliando se o seu desempenho está satisfatório. Nesse contexto foram realizadas melhorias para propiciar tais funcionalidades.

As evoluções realizadas no *plugin Work Assignment* permitem que o administrador de uma comunidade (professor), realize a modularização das atividades a serem entregues, propiciando-lhe a organização do curso e melhor visualização dos alunos.

Ao definir um módulo para seu curso, o professor informa a data de início e fim para o mesmo e, dessa maneira todas as atividades que farão parte desse módulo devem estar entre o período determinado.

Título (*)
Modulo 1

Period:
De 2015-10-20 00:00 até 2015-10-30 00:00

Imagem:
Escolher arquivo Nenhum arquivo selecionado

Rótulo da Imagem:

[+ Prévia](#) Usada quando uma versão curta do seu texto é necessária.

Descrição:

Editar Inserir Visualizar Ferramentas

Formatos Tamanho

Informações Gerais:

Figura 16 – Tela para criação de grupos de trabalhos a serem enviados.

Para criar um módulo o administrador agora pode entrar no painel de controle da comunidade e adicionar um novo tipo de conteúdo denominado “Grupo de Trabalhos a serem enviados”. Nesse conteúdo o usuário deve informar título, período, uma prévia do que será aquele módulo e sua descrição detalhada, conforme a Figura 16.

Após salvar esse grupo de trabalho o usuário é redirecionado para o módulo criado. Nesta tela é exibido ao usuário a descrição e o período referentes ao módulo. Existe

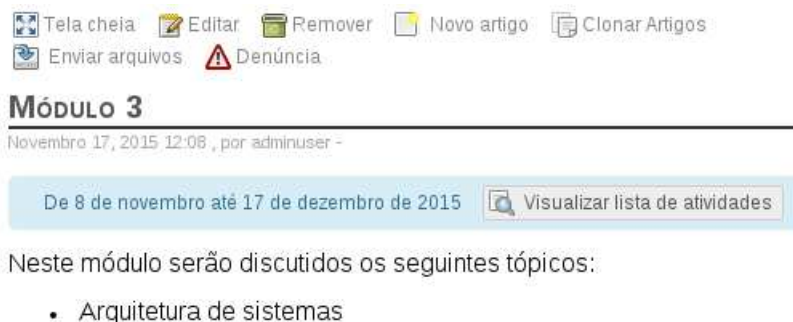


Figura 17 – Tela principal de um módulo.

ainda um botão denominado “Visualizar lista de atividades”, que mostra ao usuário ou administrador a lista de todas as atividades daquele módulo, conforme mostra a Figura 17.



Figura 18 – Lista de atividades de um *work assignment*.

Na Figura 18 é exibida a lista de atividades referentes ao módulo criado, com apenas uma atividade, a título exemplificativo. Também é exibido o *link* (“Adicionar trabalho a ser entregue”) que tem a funcionalidade de redirecionar o usuário para a página principal de criação de trabalhos.

Nesse conteúdo são solicitadas informações básicas ao usuário sobre o trabalho. Além disso é definido o período que a atividade ficará em aberto e se é possível o envio após a data limite. A Figura 19 também apresenta o gerenciamento das avaliações que permite ativar o modo de avaliação e publicação de notas de acordo com o critério de avaliação. Como evidenciado na Figura 19 é possível verificar a situação de criação de um novo “Trabalho a ser entregue”.

Como citado, uma das opções deste conteúdo é informar o tempo limite para o envio de arquivos. Buscando a melhor forma de visualização para os usuários foram criados modos de visualização e para isso definiu-se três estados de envio:

- Aberto: exhibe o tempo definido para aquela atividade e o tempo ainda restante para o término do envio (Figura 20, item 1)

Title (*)
Atividade 20

Description:
Descrição work Assignment.

Default email message:

☐ The submissions can be viewed by anyone ☐ Allow users change submissions privacy?

What is the time limit for this file to be uploaded?
From 2015-10-22 00:00 until 2015-10-22 00:00

☒ Allow the sending of files after the period

EVALUATION MANAGEMENT

☒ Activate evaluation
☒ Publish grade
Final grade options: Highest Grade

Save Save and continue

Figura 19 – Tela para criação de um *work assignment*.

Full screen Edit Delete Clone Article Upload files
Report abuse

ATIVIDADE 1
October 25, 2015 1:29, by adminuser - no comments yet

Este espaço é utilizado para a descrição do work_assignment.

① Final date to submission: **Thursday, November 19, 2015 0:00**
Remaining time for sending files: **3 days**

② Final date to submission: **Thursday, October 29, 2015 0:00**
The time limit for sending files expired, but can still be done. Delay time: **18 days**

③ Final date to submission: **Thursday, October 29, 2015 0:00**
The time limit for sending files expired! **18 days**

NO COMMENTS YET.

Post a comment

Figura 20 – Representação do estado dos trabalhos a serem entregues.

- Permitido: o usuário é informado se o período de envio já expirou e quanto tempo já se passou após a data limite (Figura 20, item 2)
- Expirado: exibe o tempo definido para a atividade e que a mesma está expirada. (Figura 20, item 3)

Para que o usuário visualize a atividade e saiba de imediato em qual situação ela se

encontra, foi utilizado um esquema de cores que fica como plano de fundo da notificação que informa o tempo restante. De acordo com a figura 20 podemos visualizar que foram utilizadas três cores: o verde para representar que a atividade está em aberto (Item 1), o laranja para informar que é permitido (Item 2), e o vermelho para indicar que está fechada (Item 3). É importante ressaltar que a Figura 20 exibe os três estados disponíveis, mas para cada “Trabalho a ser entregue” é exibido apenas um de acordo com a situação.

EVALUATION MANAGEMENT

☒ Activate evaluation

☒ Publish grade

Final grade options: Highest Grade ▼

Figura 21 – Tela para o gerenciamento de avaliações.

Ainda na tela para de criação de trabalhos a serem enviados, é importante ressaltar que para cada “Trabalho a ser enviado” o administrador pode escolher se deseja ou não publicar as notas para os alunos, permitindo ao mesmo realizar todas as avaliações antes de publicá-las.

O gerenciamento também proporciona ao administrador definir os critérios para a definição da nota final de cada atividade, conforme abaixo:

- Maior nota: é exibida como nota final a maior nota de todas as versões pontuadas.
- Mais Recente: é exibida como nota final a última versão pontuada.
- Nota opcional: no momento da avaliação de cada versão o administrador define qual é a nota final

Essas funcionalidades proporcionam flexibilidade quanto ao critério de avaliação do professor e fornecem maior liberdade para realizar as avaliações. As opções do gerenciamento de notas estão evidenciadas na Figura 21.

O professor pode pontuar os arquivos enviados pelos usuários, caso tenha habilitado a opção “Ativar Avaliação”. Como o *plugin* permite o envio de mais de uma versão para a mesma atividade, foi definido que todas elas poderiam ser pontuados. Dessa forma, a nota final é registrada de acordo com os critérios estabelecidos.

Na visualização da lista de trabalhos enviados o professor pode selecionar a opção “atribuir nota” que o redireciona para a tela que permite atribuir e alterar a nota de cada arquivo enviado, conforme evidenciado na Figura 22.

Essa tela (Figura 22) exibe ao usuário informações relevantes no momento da avaliação, a exemplo do nome do arquivo a ser avaliado, a data de envio, a data da última

Assign grade

Author name:	adminuser
File name:	(V1) golzim3.jpg
Upload at:	2015-10-27 19:12:44 UTC
Last evaluation date:	2015-10-27 17:13:33 -0200
Current Grade:	7.6
Grade	<input type="text" value="7.6"/>
<input type="button" value="Assign Grade"/> <input type="button" value="Cancel"/>	

Figura 22 – Tela para atribuição e alteração da notas.

avaliação e a nota atual (caso exista) daquele arquivo. Além disso está disponível campo para atribuição de nova nota ao aluno.

[Full screen](#)
[Edit](#)
[Delete](#)
[Clone Article](#)
[Upload files](#)
[Go back to Modulo 2](#)

[Report abuse](#)

ATIVIDADE 1

October 25, 2015 1:29 , by adminuser - no comments yet

Este espaço é utilizado para a descrição do work_assignment.

Final date to submission: **Thursday, November 19, 2015 0:00**

Remaining time for sending files: **3 days**

Author	Submission date	Versions	Grade	
adminuser	2015-11-16 02:19h	3	8.0	
(V3) uploadFile3.bt	2015-11-16 02:19h		4.0	
(V2) uploadFile2.bt	2015-11-16 02:19h		5.0	
(V1) uploadFile1.bt	2015-11-16 02:19h		8.0	

Figura 23 – Visualização das notas de trabalhos enviados.

Desde que a avaliação esteja ativa e o administrador tenha permitido a visualização das notas, é criada uma coluna que é exibe a nota final do aluno e a pontuação de cada versão avaliada. Este comportamento é evidenciado na Figura 23.

Foi implementada uma funcionalidade que tem como principal objetivo centralizar todas atividades de um usuário e deste modo organizá-las dentro de seus módulos e as comunidades pertencentes.

No painel de controle dos usuários, foi criado nova funcionalidade chamada de “Meus Cursos” que, ao ser clicada, redireciona o usuário a uma lista de todas as comuni-

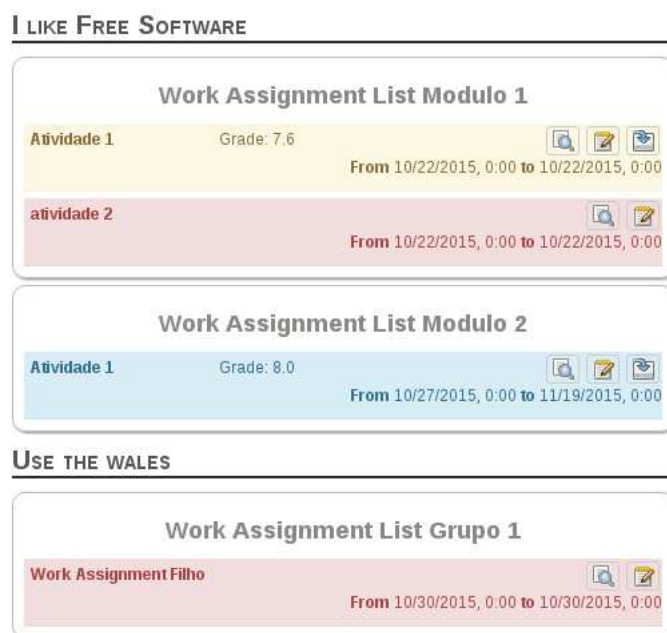


Figura 24 – Tela para visualização das atividades de forma centralizada.

dades às quais pertence e que possuem algum tipo de conteúdo relacionado ao grupo de atividades. Conforme evidenciado na Figura 24, são exibidas todas as atividades em seus respectivos grupos.

Utilizando a personalização que o Noosfero oferece, foram criados blocos que fornecem flexibilidade ao usuário quanto ao posicionamento e exibição do mesmo. Esses blocos beneficiam o usuário quanto ao *feedback* sobre suas notas. Conforme pode ser visualizado a esquerda na Figura 25, esse bloco exibe ao usuário as cinco notas mais recentes.



Figura 25 – Blocos para visualização das atividades.

Na Figura 25 evidencia-se o bloco para visualização de todos grupos e suas respectivas atividades. Por padrão são exibidas apenas três atividades por grupo, mas é

permitido ao administrador da comunidade escolher quantas deseja.

As funcionalidades deenvolvidas são importantes no contexto da UnB, pois, segundo o documento da instituição UnB (2009, p. 37), as notas para os alunos auxiliam a assimilação progressiva de conhecimento. Sabe-se, ainda que permite maior *feedback* para o aluno, pois nem sempre fica transparente o seu desempenho ao longo do semestre.

Em resumo, as funcionalidades apresentadas neste capítulo colaboram para uso do Comunidade.UnB, tornando-o uma plataforma híbrida entre um ambiente virtual de aprendizagem e uma rede social, para a troca de conhecimento através das comunidades e perfis de usuários. Assim, permite englobar departamentos, organizações e projetos específicos da Universidade.

6.2.1 Arquitetura do Plugin Work Assignment

Sabendo que o Noosfero está em constante evolução o *plugin Work Assignment* passará por melhorias durante sua vida útil. Assim, é importante entender como se onfigura a sua arquitetura após as implementações realizadas, afim de compreender quais classes compõem sua estrutura e como se relacionam.

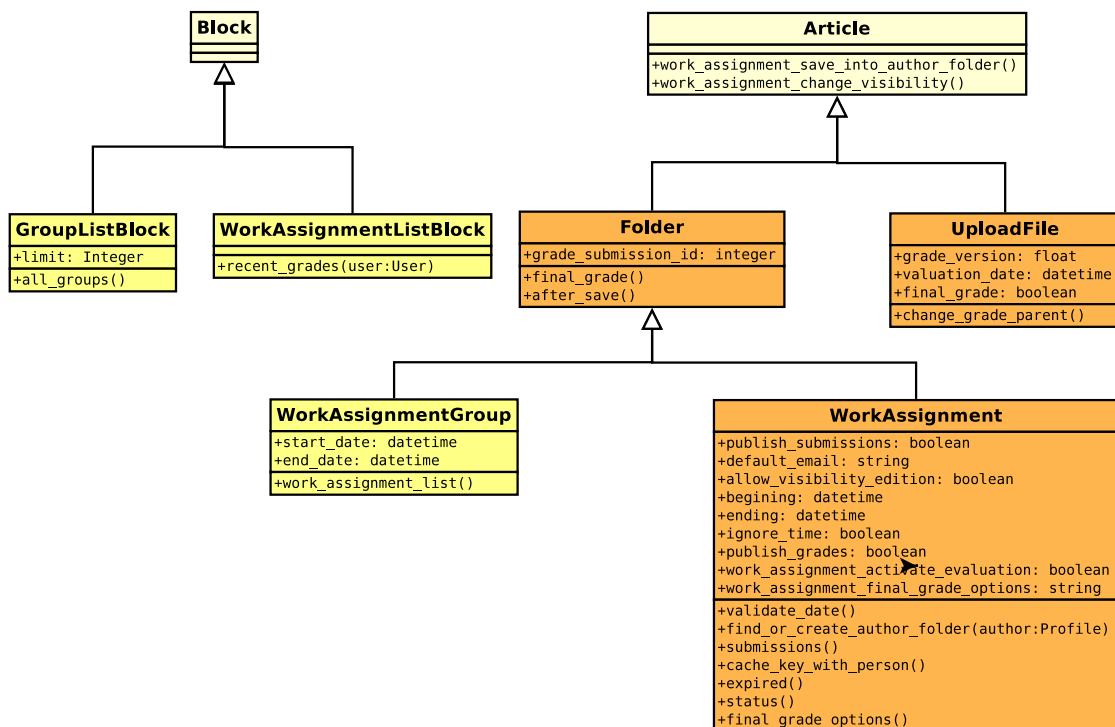


Figura 26 – Diagrama de classes do *Plugin Work Assignment*

A Figura 26 evidencia as principais classes utilizadas pelo *plugin*, no qual foram criadas as classes *GroupListBlock*, *WorkAssignmentListBlock*, *WorkAssignmentGroup*, e apenas modificadas as *WorkAssignment*, *UploadFile*, *Folder* e *Article*, *Block* que não sofreram modificações durante este processo de evolução.

No desenvolvimento foi criado uma especialização da classe *Folder* denominada como *WorkAssignmentGroup*, que possibilita ao administrador criar pastas que representem grupos de atividades e tem como principal característica a data de início e fim.

No funcionamento do *plugin*, quando um usuário cria um “Grupo de Atividades” é possível inserir qualquer tipo de *Article* dentro dessa pasta, possibilitando a criação de um “Trabalho a ser enviado” (*WorkAssignment*). A partir disso os membros da comunidade podem enviar seus arquivos como resposta a atividade, e quando o envio é realizado o *plugin* cria uma pasta (*Folder*) para cada usuário e a associa ao arquivo enviado (*UploadFile*).

Na solução foi considerado o fato de que todos os arquivos enviados por uma única pessoa poderiam ser pontuados, tornando opcional o critério para a nota final. Para isso foram criados dois atributos (*grade_version*, *valuation_date*) na classe *UploadFile*, que armazenam a nota da versão enviada e a data da avaliação, assim uma vez que a pontuação de uma versão é alterada a nota final do aluno se modifica de acordo com o critério de avaliação.

As classes *WorkAssignmentListBlock* e *GroupListBlock* são uma especialização de *Block* responsáveis por disponibilizar ao usuário a opção de blocos que exibem as notas recentes e listam todos os grupos de uma comunidade, respectivamente.

Entendendo esse funcionamento verifica-se que a arquitetura tornou o *plugin* flexível ao ponto que todas essas classe principais (*WorkAssignmentGroup*, *WorkAssignment*, *UploadFile*, *Folder*) podem ser criadas de forma independente. A dependência está relacionada aos blocos que são dispensáveis caso não existam trabalhos a serem enviados ou grupos de atividades.

6.3 A evolução do *plugin Comunidade.UnB*

Como o *plugin* permitia que apenas os novos usuários realizassem o cadastro de suas matrículas, foi criada nova versão que permite cadastrar a matrícula dos usuários que já possuem registro no Comunidade.UnB. Dessa maneira, o *plugin Comunidade.UnB* é ativado e permite a autenticação apenas dos usuários cadastrados no LDAP da universidade. Esta base de dados contempla alunos, professores, servidores técnico-administrativos que queiram utilizar a rede. A história de usuário (US09) com seus respectivos cenários de uso, encontram-se disponíveis no Seção 5.3.

Com a nova funcionalidade após o usuário realizar o Login, ele é redirecionado para uma tela que solicita a matrícula e senhas do Portal e do LDAP, conforme a Figura 27. O sistema verifica os dados na base de dados LDAP e caso seja confirmado, cadastra matrícula e altera a senha do usuário.

A partir de agora a autenticação no Comunidade.UnB é realizada de acordo com a base dados LDAP da Universidade de Brasília. Informe sua matrícula e senhas para acessar o portal.

ALTERAR SENHA

Matricula

Senha atual

LDAP password

Alterar senha

Figura 27 – Tela para associar matricula ao usuário.

Foi retirada a funcionalidade de alteração e recuperação de senha uma vez que queremos manter a compatibilidade entre a rede Comunidade.UnB e os demais serviços. Desta forma, caso o usuário deseje trocar sua senha, o mesmo deve procurar o CPD e solicitar a alteração.

Com o objetivo de utilizar este *plugin* em um ambiente de produção é necessário solicitar acesso ao LDAP de alunos do matricula web, visto que o da FGA segue um mesmo padrão de senha para todos os usuários, dessa forma todos ficariam com suas contas vulneráveis à acessos indevidos.

Como visto o *plugin* traz para o Comunidade.UnB maior controle sobre seus usuários, centralizando o acesso para o escopo da Universidade.

7 Conclusão

No decorrer deste trabalho de conclusão de curso, foram realizadas atividades que englobam as principais áreas de conhecimento da Engenharia de Software, desde o levantamento de requisitos à implementação e implantação das funcionalidades desenvolvidas. Houve contribuições no contexto da manutenção e evolução de uma plataforma real de software livre, onde foi priorizado a evolução do conhecimento de forma colaborativa com os membros do laboratório avançado de pesquisa e produção de software.

O desenvolvimento proporcionou a participação em um processo distribuído e colaborativo de desenvolvimento de software, dado que o desenvolvimento foi realizado em conjunto com uma comunidade de software livre com equipes espalhadas pelo país.

No trabalho foram identificadas e detalhadas as principais funcionalidades de um ambiente virtual de aprendizagem comparando-as com as disponíveis no Noosfero. Dessa maneira foram selecionadas as mais importantes para os usuários, implementadas e integradas a rede Comunidade.UnB.

A rede colaborativa Comunidade.UnB foi atualizada para uma versão mais recente que provém de novas funcionalidades e melhores níveis de segurança, além de habilitar o protocolo de segurança SSL para manter a integridade das informações dos usuários. Com evolução dos *plugins* a autenticação, em ambiente de testes, é realizada de acordo com a base de dados LDAP e agrega ao professor funcionalidades que proporcionam melhor controle das atividades disponibilizadas e a atribuição de notas.

Portanto a Rede Comunidade.UnB está atualizada com a última versão de pacotes do Noosfero, Dessa maneira espera-se que o Comunidade.UnB transforme-se em um ambiente híbrido para os alunos e professores compartilharem o conhecimento de maneira construtiva e evolutiva.

7.1 Limitações

Ao longo deste trabalho foi realizado a evolução do *plugin* Comunidade UnB que tem como objetivo prover a autenticação via LDAP. Apesar dos testes terem sido realizados no ambiente de homologação torna-se inviável colocá-lo em produção, porque a base de dados LDAP fornecida pelo CPD da FGA é a mesma utilizada para o acesso a rede *wireless* e segue um mesmo padrão de senhas para todos alunos. Caso seja colocado em produção, todos usuários ficariam com problemas de segurança dado que não é possível alterar a senha na base LDAP. Como solução deve ser solicitado acesso a base de dados única da Universidade de Brasília, que contém todos os alunos que utilizam o sistema de

matrícula, visto que os testes comprovam que o *plugin* está preparado para isso.

7.2 Trabalhos Futuros

Nesta seção é tratado a perspectiva de continuidade na evolução da plataforma Noosfero no contexto abordado neste trabalho, dessa maneira foi feito um levantamento das principais histórias de usuário para inspirar os próximos colaboradores.

Evoluir calendário de atividades

O Noosfero dispõe de um calendário para permitir a visualização dos eventos de maneira organizada. Assim sendo é proposto melhorias para que o usuário visualize todas as atividades criadas pelo professor de acordo com a data de entrega.

Histórias de usuário

1. Visualizar atividades no calendário

Como um usuário

Gostaria de visualizar todas as atividades propostas pelo professor

Para me inteirar sobre todas as atividades pendentes.

Cenários de uso:

- a) Integrante visualiza atividades marcadas
- b) Visualizar detalhes das atividades

Mecanismos de notificações

1. Notificações por email

Como um usuário

Gostaria de receber notificações no meu email

Para ser alertado sobre as atividades.

Cenários de uso:

- a) Permitir que o usuário defina se deseja ser notificado via email
- b) Permitir notificações quando tempo de atividade vai expirar
- c) Notificar via email quando nova atividade for criada

Ferramentas para notas

1. Ponderar notas

Como um professor

Gostaria de ponderar as notas dos módulos da disciplina

Para definir a menção dos alunos.

Cenários de uso:

- a) Definir quais os módulos fazem parte da média do aluno
- b) Selecionar quais as atividades fazem parte da avaliação
- c) Criar média ponderada para as atividades
- d) Exportar médias em formato CSV

Referências

- AGUADO, A. G. *Dimensões de qualidade em Ambientes Virtuais de Aprendizagem*. Tese (Doutorado) — Universidade de São Paulo, 2013. Citado na página 32.
- ARAÚJO, T. C. *AppRecommender: um recomendador de aplicativos GNU/Linux*. Tese (Doutorado) — Universidade de São Paulo, 2011. Citado na página 21.
- BECK, K. et al. *The agile manifesto*. 2001. Citado na página 28.
- BEHAR, P. A.; BASSANI, P. Avaliação da aprendizagem em ambientes virtuais. *Modelos pedagógicos em educação a distância*. *Puerto Alegre: Artmed*, p. 93–113, 2009. Citado na página 31.
- BERNARDO, P. C. *Padrões de testes automatizados*. Tese (Doutorado) — Universidade de São Paulo, 2011. Citado na página 30.
- BLACKBOARD, G. A. *Sobre a Bb - Construindo a Melhor Experiência Educacional*. 2015. Disponível em: <<http://blackboard.grupoa.com.br/sobre/sobre-a-bb>>. Acesso em: 11 de Junho de 2015. Citado na página 45.
- BUCHER, D. C. Rede de colaboração social para universidades brasileiras: um estudo de caso de implantação e desenvolvimento distribuído de uma plataforma livre na universidade de Brasília. 2013. Citado 6 vezes nas páginas 33, 39, 40, 41, 42 e 56.
- CASTELLS, M. A era da intercomunicação. *Caminhos para uma comunicação democrática*. São Paulo, Instituto Paulo Freire, 2007. Citado na página 21.
- CHIN, P. *Virtual Learning Environments*. [S.l.]: Learning and Teaching Support Network, 2003. Citado na página 32.
- CORBUCCI, H. *Métodos ágeis e software livre: um estudo da relação entre estas duas comunidades*. Tese (Doutorado) — Universidade de São Paulo, 2011. Citado 2 vezes nas páginas 29 e 30.
- COSTA, L. B. da. *Redes: uma introdução às dinâmicas da conectividade e da auto-organização*. [S.l.]: WWF-Brasil, 2004. Citado na página 33.
- CRISCUOLO, M. *Qualidade de Produto de Software: uma abordagem baseada no controle da complexidade*. Tese (Doutorado) — Universidade de São Paulo, 2008. Citado na página 26.
- DAVIS, C. H. et al. Social media in higher education: A literature review and research directions. University of Arizona and Claremont Graduate University, 2012. Citado 2 vezes nas páginas 22 e 33.
- DUTRA, R. L. de S.; TAROUÇO, L. M. R. Objetos de aprendizagem: uma comparação entre scorm e ims learning design. *RENOTE*, v. 4, n. 1, 2006. Citado na página 47.
- DUVALL, P. M.; MATYAS, S.; GLOVER, A. *Continuous integration: improving software quality and reducing risk*. [S.l.]: Pearson Education, 2007. Citado na página 30.

- ELLISON, N. B. et al. Social network sites: Definition, history, and scholarship. *Journal of Computer-Mediated Communication*, Wiley Online Library, v. 13, n. 1, p. 210–230, 2007. Citado na página 33.
- EMIRBAYER, M.; GOODWIN, J. Network analysis, culture, and the problem of agency. *American journal of sociology*, JSTOR, p. 1411–1454, 1994. Citado na página 33.
- GNU, O. S. *O que é software livre? @ONLINE*. 2013. Disponível em: <<http://www.gnu.org/philosophy/free-sw.en.html>>. Citado na página 27.
- IEEE 1219. *Std 1219 – IEEE Standard for Software Maintenance*. New York, NY, USA, 1998. Citado 2 vezes nas páginas 25 e 26.
- ISO/IEC. *Systems and software engineering — Software life cycle processes*. [S.l.], 1998. Citado na página 25.
- LARMAN, C. *Utilizando UML e padrões*. [S.l.]: Bookman, 2002. Citado na página 39.
- LIENTZ, B. P.; SWANSON, E. B. Software maintenance management. *Addison—Wesley*, 1980. Citado na página 26.
- MARTELETO, R. M. Análise de redes sociais: aplicação nos estudos de transferência da informação. *Ciência da informação*, SciELO Brasil, v. 30, n. 1, p. 71–81, 2001. Citado na página 33.
- MEIRELLES, P. R. M. *Monitoramento de métricas de código-fonte em projetos de software livre*. Tese (Doutorado) — Universidade de São Paulo, 2013. Citado 3 vezes nas páginas 27, 29 e 35.
- MESZAROS, G. *xUnit test patterns: Refactoring test code*. [S.l.]: Pearson Education, 2007. Citado na página 30.
- MYERS, G. J.; SANDLER, C.; BADGETT, T. *The art of software testing*. [S.l.]: John Wiley & Sons, 2011. Citado na página 30.
- NORTH, D. Introducing behaviour driven development. *Better Software*, 2006. Citado na página 37.
- PFLEEGER, S. L. Engenharia de software: teoria e prática. 2ª Edição, *Prentice Hall*, 2004. Citado 2 vezes nas páginas 25 e 26.
- PRESSMAN, R. S. *Engenharia de software*. [S.l.]: McGraw Hill Brasil, 2011. Citado na página 26.
- RAYMOND, E. S. *The Cathedral & the Bazaar*. Sebastopol, CA, USA: O'Reilly & Associates, Inc., 1999. Citado na página 28.
- REIS, C.; FORTES, R. P. de M. Caracterização de um modelo de processo para projetos de software livre. *São Carlos, SP*, 2001. Citado na página 30.
- ROCHA, H. V. d.; MORAES, M. O ambiente teleduc para educação à distância baseada na web: Princípios, funcionalidades e perspectivas de desenvolvimento. *Educação a distância: Fundamentos e práticas*. Campinas, SP: UNICAMP/NIED, p. 197–212, 2002. Citado na página 46.

SANTOS, A. P. O. dos. *Aplicação de práticas de usabilidade ágil em software livre*. Tese (Doutorado) — Universidade de São Paulo, 2012. Citado na página 28.

SCHLEMMER, E. Ambiente virtual de aprendizagem (ava): uma proposta para a sociedade em rede na cultura da aprendizagem. *Aprendizagem em ambientes virtuais: compartilhando ideias e construindo cenários*. Caxias do Sul, RS: Educs, 2005. Citado na página 31.

SJOBERG, D. I.; DYBA, T.; JORGENSEN, M. The future of empirical methods in software engineering research. In: IEEE COMPUTER SOCIETY. *2007 Future of Software Engineering*. [S.l.], 2007. p. 358–378. Citado na página 26.

STALLMAN, R. M. What is free software. *Free Society: Selected Essays of*, 2002. Citado na página 27.

STALLMAN, R. M.; GAY, J. *Free software, free society: Selected essays of Richard M. Stallman*. [S.l.]: CreateSpace, 2009. Citado na página 28.

TOMAÉL, M. I.; ALCARÁ, A. R.; CHIARA, I. G. D. Das redes sociais à inovação. *Ciência da informação, Brasília*, SciELO Brasil, v. 34, n. 2, p. 93–104, 2005. Citado na página 33.

UNB. *Guia do Professor*. 2009. Disponível em: <<http://www.unb.br/noticias/downloads/guiadoprofessorfinal.pdf>>. Acesso em: 18 de Junho de 2015. Citado na página 75.

Apêndices

APÊNDICE A – Instruções para preenchimento do questionário

Caros,

O estudo a seguir faz parte de uma pesquisa realizada por um aluno de Engenharia de *software*, da Universidade de Brasília – **UnB**, o qual será utilizado como forma de obtenção de conhecimento sobre o tema.

Esta pesquisa visa investigar se os alunos da Universidade de Brasília utilizam as redes sociais, como o *Facebook*, para o compartilhamento de recursos e informações relacionados as disciplinas cursadas na Universidade. Para tanto, necessito de sua colaboração respondendo ao questionário sobre o tema. É importante ressaltar que os dados aqui levantados serão compilados e utilizados apenas para fins acadêmicos, na problematização da pesquisa.

Instruções para preenchimento do questionário:

1. Você deverá selecionar um item para indicar o grau de concordância com as assertivas apresentadas na tabela 6.

NUNCA	RARAMENTE	ÀS VEZES	FREQUENTEMENTE	SEMPRE
-------	-----------	----------	----------------	--------

Tabela 6 – Opções para preenchimento do questionário.

Lembrando que quanto mais próximo de **Nunca**, **MENOR** o grau de concordância. Quanto mais próximo de **Sempre**, **MAIOR** o grau de concordância.

2. O questionário deve ser respondido pelos alunos das Universidade de Brasília;
3. Não existem respostas “certas” ou “erradas”. O importante é respondê-las de forma **sincera**.
4. Procure não deixar nenhuma resposta em branco. Sua participação é muito importante para a finalização deste trabalho.

Desde já agradeço sua atenção e cooperação. Coloco-me à disposição para os esclarecimentos que se fizerem necessários.

Cordialmente,

Hebert Douglas de Almeida Santos <hebertdoug@gmail.com>

APÊNDICE B – Questionário para problematização

1. Com que frequência você utiliza redes sociais?
2. Você utiliza alguma rede social como ferramenta de apoio as disciplinas?
3. Os professores incentivam o uso de redes sociais para suas disciplinas?
4. Mesmo que o professor não recomende o uso de redes sociais para discussão de conteúdos de suas disciplinas, você as utiliza?